

# Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource Vol. 3 No. 9 Sept. 1988  
US \$3.95 Canada \$4.95

## TEACHING WITH THE AMIGA

TAKE A LOOK AT WHAT'S IN  
THIS MONTH'S AC!

- COMPUTER AIDED INSTRUCTION  
IN AMIGABASIC
- KIDEOS BY ELEMENTARY  
STUDENTS
- SHAKESPEARE REVIEW
- GELS IN MULTI-FORTH, PART II
- INSIDE X-SPECS 3D

ALSO: COMPLETE COVERAGE OF  
AMIEXPO MIDWEST

AmigaBASIC - Inside and Out

Charles Berg, Kathleen Troth INVESTIGATING MUSIC

DESKTOP VIDEO: A GUIDE TO PERSONAL AND  
SMALL BUSINESS VIDEO PRODUCTION

MATHEMATICS

FINITE

IMAGES

SCIENCE

Communicating

ALGEBRA 1: a modern course

INTELLITYPE

WAIT! THERE'S MORE!

BEETHOVEN: SYMPHONY NO. 9  
THE CLEVELAND ORCHESTRA  
GEORGE SELL

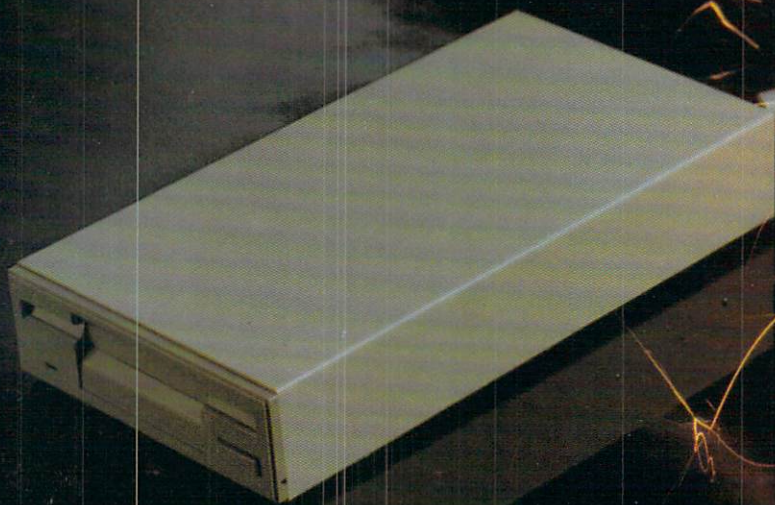
1084

Commodore





# Our DRIVE is Dynamite!



## The AIR DRIVE

Dual-Sided, 880K

Color-coordinated chassis and faceplate

Smaller and lighter than the Amiga drive

External drive pass-through

Extra long 30" interface cable

Spring-loaded dust door

Compatible with all Amiga systems

One year warranty

Internal drive also available

**CALL FOR DEALER NEAREST YOU.**



**\$79** Small and powerful.  
The Avatex 1200E.

## Avatex MODEMS

\$105.....	1200P
\$99.....	1200HC
\$169.....	2400
\$69.....	1200i
\$159.....	2400i

## FREE WITH EACH MODEM

Amiga communication software  
& CompuServe access time.

## MEGATRONICS



To Order: 800-232-6342

Ask about our  
**LOWEST  
PRICE  
GUARANTEE**

INSIDE UTAH: (801) 752-2642  
FAX: (801) 752-8752  
CALL FOR A FREE CATALOG.  
CREDIT CARDS VERIFIED  
FOR YOUR PROTECTION.

MEGATRONICS, INC. BOX 3660, LOGAN, UT 84321





***AT LAST!***

***...real-time, LIVE! video on your  
Amiga's screen.***

- ***True Color:*** just as it comes from your video source: camera, VCR, TV, anything. Direct, moving, in your Amiga's memory...our patented technology.
- ***Fast:*** video images in black & white, 32-color, and 4,096-color HAM. See 15 new images every second in black & white, 12 in color, 4 in HAM.
- ***Save:*** moving video, play it back, use it in other programs. Unlimited stills, too.
- ***Video Effects:*** real-time mouse-controlled...posterization, fades, color-keying, strobos, more.
- ***Roll Your Own:*** programmer's video library, hardware documentation, examples in C, Basic.

*For more information, contact:*

***$\alpha^2$***

**A-Squared Distributions Inc.**

6114 La Salle Avenue, Suite 326  
Oakland, California 94611  
415-339-0339



## Amazing Features

- |  |    |   |    |
|--|----|---|----|
| <b>The Kideo Tapes</b> by John Dandurand<br>A Georgia elementary school puts desktop video to work.                              | 15 | <b>Computer Aided Instruction</b> by Paul Castonguay<br>Create your own lessons with this generalized authoring system in AmigaBASIC.             | 71 |
| <b>Speeding Up Your System</b> by Tony Preston<br>Fuel inject your system with floppy disk caching, not an expensive hard drive. | 41 | <b>Gels in Multi-Forth, Part II: Screenplay</b> by John Bushakra<br>Make the IFF converter from Part I easy to use with gadgets, menus, and more. | 83 |
| <b>Amiga Product Guide: Education Edition</b><br>Everything you need to send your Amiga to the head of the class.                | 49 | <b>AmiExpo Midwest '88</b> by Michael T. Cabral<br>After taking the coasts by storm, the Amiga wows the Midwest.                                  | 91 |

## Amazing Reviews

- |   |   |   |    |  |    |
|---|---|---|----|--|----|
| <b>Intellitype</b> by Harv Laser<br>Learning to type made easy...and <i>fun</i> ? | 8 | <b>Shakespeare</b> by Barney Schwartz<br>Bring your desktop publishing to life with full color. | 10 | <b>XSpecs 3D</b> by Steve Hull<br>A new dimension in Amiga graphics. | 20 |
|---|---|---|----|--|----|

## Amazing Columns

- |   |    |  |     |
|---|----|--|-----|
| <b>AmigaNotes</b><br>by Richard Rae<br>Ever wonder how IFF sound samples are stored?                            | 23 | <b>Hot on the Shelves</b><br>by Michael T. Cabral & Michael Creeden<br>What do you get when you combine intense war strategy with a monochrome monitor and desktop presentation? Check it out. | 29  |
| <b>Take Five!</b><br>by Steve Hull<br>Beat the back-to-school blues with five more sizzling summer game titles! | 27 | <b>Bug Bytes</b> by John Steiner<br>Chillier weather, but still plenty of bugs and upgrades.   | 39  |
| <b>The Command Line</b><br>by Rich Falconburg<br>The next stop on a continuing tour of the CLI.                 | 45 | <b>C Notes from the C Group</b> by Stephen Kemp<br>Operators, expressions, and statements in C uncovered.  | 65  |
|   |    | <b>Roomers</b> by The Bandito<br>Can an Apple IIgs Plus a day keep the Amiga away?   | 102 |

## Amazing Departments

- |              |   |  |    |                                |     |
|--------------|---|--|----|--------------------------------|-----|
| Amazing Mail | 4 | Index of Advertisers/<br>Reader Service Card | 96 | Public Domain Software Catalog | 105 |
|--------------|---|--|----|--------------------------------|-----|



The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™. If you are not an Amazing Dealer, but would like to become one, call PiM Publications, Inc.:

[illegible]

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, Software Etc., selected WaldenBooks Stores, and Walden's Software Store locations.



## Including the .bmap in your BASIC Program!

Dear Amazing Computing:

If you program with AmigaBASIC using library calls, the appropriate '.bmap' file must be present along with your program. It seemed untidy to me to have this separate file. If you use the Absoft compiler, then it defeats the objective of a single stand-alone program file. Here is a way to include the bmap file in your BASIC program. First, use a program like this to convert the bmap file to a sequential file of ASCII hex data that can be merged with your Basic program:

```
INPUT "File name ";file$          'the bmap file
OPEN file$ FOR INPUT AS #1
  Flength=LOF(1):Q$=CHR$(34)      'use chr$(34) to allow double quotes
  INPUT "Destination file name ";Destfile$
  OPEN Destfile$ FOR OUTPUT AS #2
  PRINT #2, "REM - Conversion of ";file$
  PRINT #2, "OPEN ";Q$; "RAM:" +file$;Q$ FOR OUTPUT AS #1"
  PRINT #2, "FOR X=1 TO ";Flength;: :READ X$:A=VAL("Q$+"&HO"+Q$+X$) "
  PRINT #2, "PRINT #1, CHR$(A);"
  PRINT #2, "NEXT X"
  PRINT #2, "CLOSE #1,"
  WHILE count<=(Flength)
    IF count MOD 20 = 0 THEN
      PRINT #2, " "
      PRINT #2, "DATA";
    ELSE
      PRINT #2, ",";
    END IF
    count=count+1
    PRINT #2, HEX$(ASC(INPUT$(1,1)));
  WEND
  PRINT #2, "": CLOSE #2:CLOSE #1
PRINT "Merge "Destfile$;" with program that needs bmap"
```

This is a typical sequential file produced:

```
REM _ Conversion of dos. bmap
OPEN "RAM:dos.bmap" FOR OUTPUT AS #1
  FOR X=1 TO 433 :READ X$:A=VAL("Q$+"&HO"+X$)
  PRINT #1, CHR$(A);
  NEXT X
CLOSE #1
DATA 78, 4F, 70, 65, 6E, 0, FF, E2, 2, 3, 0, 78, 43, 6C, 6F, etc...
```

Next, merge this with your program and add your library call after the above routine has recreated your bmap file on the Ram disk.

eg:

LIBRARY "RAM:dos.library"

etc.

Write your program so that the bmap file(s) are deleted upon termination of the program. (I told you I like things tidy!)

Regards,  
Frank Turner  
California

## ASCII to ProWrite Through The Back Door

Dear Amazing Computing:

Thanks for the great review of ProWrite in Vol 3.7. Since I have been using the program, I have discovered a feature (?) which neither the manual nor your review covered.

If you've ever tried to open an ASCII text file using ProWrite's "open" requester, you will notice that only files saved in ProWrite format appear on the disk directories. The ProWrite manual incorrectly states that only ProWrite files can be loaded into ProWrite. There are, however, two ways of loading ASCII files into ProWrite: (continued on page 6)

# Amazing COMPUTING

## ADMINISTRATION

**Publisher:** Joyce Hicks  
**Assistant Publisher:** Robert J. Hicks  
**Circulation Manager:** Doris Gamble  
**Asst. Circulation Mgr:** Traci Desmarais  
**Corporate Trainer:** Virginia Terry Hicks  
**Traffic Manager:** Robert Gamble  
**International Coordinator:** Marie A. Raymond

## EDITORIAL

**Managing Editor:** Don Hicks  
**Co-Editor:** Ernest P. Viveiros Jr.  
**Co-Editor:** Michael T. Cabral  
**Hardware Editor:** Ernest P. Viveiros Sr.  
**Music & Sound Editor:** Richard Rae  
**Copy Editors:** Julie Landry  
Michael Creeden

## PRODUCTION

**Art Director:** Keith Conforti  
**Illustrator:** Brian Fox  
**Production Manager:** Rico Conforti  
**Associate Prod. Mgr:** Mark Thibault

## ADVERTISING SALES

**Advertising Manager:** John D. Fastino

1-508-678-4200

FAX 1-508-675-6002

## SPECIAL THANKS TO:

Buddy Terrell & Byrd Press  
Betsy Piper at Tech Plus  
Bob at Riverside Art, Ltd.  
Swansea One Hour Photo  
Boston Jewelry & Loan of Fall River

Amazing Computing™ (ISSN 0886-9480) is published monthly by PiM Publications, Inc., Currant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for \$24.00; in Canada & Mexico surface, \$36.00; foreign surface for \$44.00.

Application to Mail at Second-Class Postage Rates pending at Fall River, MA and additional mailing offices.

**POSTMASTER:** Send address changes to PiM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A. Copyright © Aug 1988 by PiM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PiM Publications, Inc. maintains the right to refuse any advertising.

PiM Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format to the Co-Editor. Requests for Author's Guides should be directed to the address listed above.





#### Other Products from The Other Guys

**REASON** - a professional proofreading system used by universities and writers around the world to analyze and improve writing. (Has helped raise students grades when used faithfully.) \$395.00

**OMEGA FILE** - a *REAL* data base & mail merge \$79.99

**PROMISE** - the *BEST* high speed spell checker. (Even better than Zing!® Spell) \$49.99

**KEEP-Trak GL** - general ledger for home or business \$49.99

**AMT** - amortization program \$39.99

**MATCH-IT** - teaches shapes & colors (preschool) \$39.99

**MATH-A-MAGICIAN** - add, subtract, multiply & divide \$39.99

Call or write for more information.

## **SYNTHIA** High Performance Digital Synthesizer

A state of the art music tool which will:

Create digital **IFF Instruments** for use with nearly all music programs!

Modifying existing **IFF Instruments**. Use **SYNTHIA** on digitized samples to add reverb, wow, and other enhancements.

#### **SOMETHING FOR EVERYONE:**

**Additive Synthesis** - a traditional method which can create almost any type of instrument.

**Plucked String Synthesis** - simulates plucked strings . . . right down to the 'pluck'.

**Interpolative Synthesis** - a method which introduces the natural imperfections found in instruments. (Instruments such as brass, woodwinds, pianos, etc.)

**Percussion** - build your own drum set . . . create any drum you desire.

**Subtractive Synthesis** - a simple method of creating instruments.

**Special Effects** - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .

**IFF Music Player** - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

#### **IS IT LIVE . . . OR IS IT SYNTHIA?**

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!

Why buy digitized instruments when you can **SYNTHIAsize** them?

Requires AMIGA 512K

**\$99.99**

Copyright©1987, THE OTHER GUYS Software • AMIGA is a registered trademark of Commodore Amiga

**NOW SHIPPING**



**THE OTHER GUYS**

55 North Main Street  
Suite 301-D  
PO Box H  
Logan Utah 84321

(801) 753-7620  
**(800) 942-9402**





(Amazing Mail continued from page 4)

The CLI way.

```
1>ProWrite [.MR] (SPACE) (FILENAME)
```

(The filename can also include a directory path)

The other way.

Copy the .info file from another ProWrite document to the file you need to use. Now, simply double click on the newly-created icon. That's It!!

Darrin Raposa  
Massachusetts

## Where Are The Doubly-Linked Lists?

Dear AC:

I wish William Gammill had mentioned one other thing about linked lists in C (Amazing Computing, V3.7). Namely, a set of functions for handling doubly-linked lists is an integral part of the Amiga's Exec!

There are two types of doubly-linked lists supported by AmigaDOS 1.2 and later—the List, which anchors a set of Nodes, and the MinList, which anchors a set of MinNodes. The difference between them is that MinNodes lack the priority and name pointer fields (In\_Pri, In\_Name), so the FindName and Enqueue functions don't work on them, but they require less memory. The In\_Type field is also not present for MinNodes, but none of the standard list functions use it. Its primary use is as an identifier for debugging and error-detection purposes, and user-written programs may do as they wish with it.

In general, I recommend using the Exec list functions where possible. There's nothing magical about them; they're already debugged and documented (and the ones that aren't macros are still taking up memory whether you use them or not), and you can easily piggyback any special capabilities on top of them, just like Commodore did for their Message and I/O systems, etc.—in object-oriented programming terms, this is known as inheritance.

After all—how many automobile manufacturers make their own tires?

Sincerely,  
Tim Holloway  
Florida

[Look for doubly-linked list techniques in  
**"More Linked Lists in C,"** coming soon  
in *Amazing Computing*.—Ed]

## Graphics the Easy Way!

Dear Amazing Computing:

A hint for other non-artists. Get patterns for cross stitch from yarn shops or department stores. Pick a square brush and follow the pattern; it's extra easy in the magnifying mode. Patterns range from simple to intricate. It's inexpensive clip art.

Yours Truly,  
Charles Clyde  
Pennsylvania

## More Amiga User Groups!!!!

Amazing Computing:

The Sacramento Amiga Computer Club boasts 300 members and meets monthly. In addition to holding several Special Interest Group (SIGS) meetings, we produce a monthly club disk and newsletter, the *AMIGAZette*. SACC is interested in communicating with other User Groups by exchanging Newsletters.

Sacramento Amiga Computer Club  
ATTN: Amigazette Editor  
P.O. Box 19784  
Sacramento, CA 95819-0784

Robert Du Gaue  
AMIGAZette Editor

Dear Amazing Computing  
Although we have been meeting for several months, it was not until this month we became a formal organization. Our name and meeting address is:

Border States Amiga Users Group  
1614 Towson Ave  
Fort Smith, AR 72901  
Phone 782-4048

We meet on the 2nd Saturday of each month from 9am - noon.

Our officers are :

Wayne Pace - President  
Wayne Hyman - Secretary  
Bill Webb - Librarian

Our present contact person and mailing address is :

Wayne Hyman  
1501 Independence  
Fort Smith, AR 72901  
Phone (501) 646-9582

Dear Amazing Computing:

The Quad Cities Amiga Club holds meetings on the first Tuesday (Business Meeting and Software/Hardware Demonstrations) and the third Thursday (General Information Sharing) of each month and sponsors a local Amiga BBS (309-762-4980) operating 24 hours a day 300/1200/2400 baud with (B)ulletin, (M)essage, (E)mail and library sections (Uploads Graciously Accepted).

Since our club now includes about 50 owners of Amiga computers, we are planning to print a newsletter on a regular basis. Anyone interested in our group may contact our BBS by modem or any officer by phone:

Greg Dengler President 319-323-2261  
Mike Asplund V.P. 309-764-6512  
Sheila Jackson Secretary 309-797-5335  
Doug Sample Librarian 309-944-3481  
Rita Tank Editor 309-792-1603

or by mail at:

Quad Cities Amiga Club  
c/o Michael Nerdahl  
2418 - 15th Avenue  
Moline, IL 61265

We would be interested in hearing from other Amiga user groups. Thanks for your interest and support to user groups such as ours.

Sincerely,  
Michael Nerdahl  
Corresponding Secretary

Dear AC:

Please enter us in your database of user groups:

Amiga Users' Group  
Fort Devens Computer Club  
137 Walnut Street  
Fort Devens, MA 01433

Don Christensen President  
John Posco Vice President  
Dave Baum Librarian  
BBS number: 508-772-7322

Thanks for your interest in user groups!

Yours,  
Don Christensen

### We welcome your comments!

All AC readers who have letters, questions, or comments printed in AC receive a certificate good for 5 free Public Domain Software disks  
Keep involved, Please write us!



A new release from Lattice  
who brought you  
the better C:

IT  
CAME  
FROM  
THE  
C

## STARRING LATTICE C++!

- ★ **OBJECT ORIENTED.** A FIRST FOR YOUR OBJECT ORIENTED AMIGA!
- ★ **LIFE** SPRINGS FROM YOUR SCREEN WITH THIS HIGHER LEVEL OF PROGRAMMING!
- ★ **PRODUCE** MORE RELIABLE PROGRAMS WITH FEWER KEYSTROKES & LESS EFFORT!
- ★ **HIDE** YOUR DATA TO PROTECT IT!
- ★ **BUILD** ONE TYPE ON ANOTHER TYPE!
- ★ **DEFINE** A GENERIC THEN REDESIGN SPECIFICS WITHOUT MAKING THE PROGRAM FAIL!
- ★ **TRUE** DATA ABSTRACTION TO AID IN PORTABILITY AND PROGRAM SIMPLICITY!
- ★ **OVERLOAD** OPERATORS AND FUNCTIONS! COMPLEX ARITHMETIC LOOKS LIKE PART OF THE LANGUAGE!

**Now at a software supplier  
near you!**

  
**Lattice**

Subsidiary of SAS Institute Inc.

Lattice, Incorporated  
2500 S. Highland Avenue  
Lombard, IL 60148  
Phone: 800/533-3577  
In Illinois: 312/916-1600

Lattice is a registered trademark of Lattice, Incorporated.  
Amiga is a trademark of Commodore-Amiga, Inc.



# IntelliType

by Harv Laser

## Is A Non-Typist Handicapped?

Jim is a close friend of mine; we've known each other for almost 20 years. Like me, Jim is a techno-freak. He loves new electronic toys. He always has the latest stereo and video gear, reads all the magazines, and talks about consumer technology constantly. But Jim knows very little about computers except that he wants one. He's watched me using my Amiga many times, and it's easy to impress him with the things it can do. Jim hasn't bought a computer because he never learned to type. Lacking the ability to type in this age of computers embarrasses him. He knows the Amiga has this terrific interface with a mouse and icons, but he also knows that without the ability to touch type, he is hampered in what he can do with the machine. Not being able to tap all this electronic wizardry is very frustrating for him.

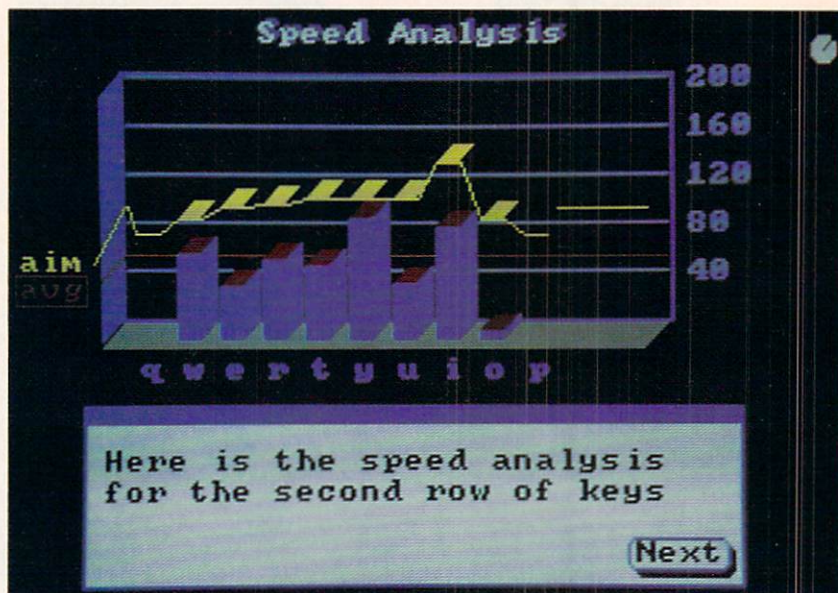
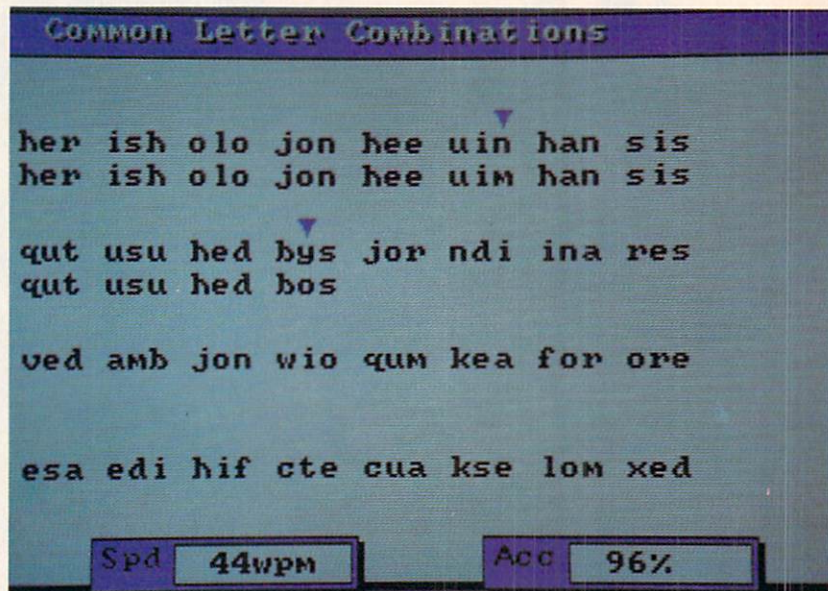
Back in the days when Jim and I went to high school, taking a typing class was not something people bragged about. When you're a teenager, learning how to type sounds about as exciting as learning how to boil water or weave baskets. It's boring and repetitious and it won't help you on dates. Successful business people have secretaries to do the typing anyway, right?

I took a semester of typing in high school because my parents said it would really help me later in life. Sometimes it pays to listen to the old folks when you're a kid. Jim never took a typing class and now, twenty years later, he knows he made a mistake.

When he's at work, he must either write up his reports and beg someone to type them, or sit for hours using the old "hunt and peck" two-finger method. The next time Jim's over for a visit, I think I'll show him IntelliType. For Jim and other people who want to type and have access to an Amiga, I can't think of a better way to learn.

## A Better Way To Learn

Electronic Arts' IntelliType is a new typing tutor program for adults. It's not one of those old "zap the aliens with letters" programs, but it's not dry and boring either. It's simply the best computerized typing teacher I've ever seen.



(continued on page 34)



# AmiEXPO

**October 7-9, 1988**  
**Westin Bonaventure**  
**Los Angeles, California**  
**The Amiga Event!**

## Exhibition Hall

Over 100 Amiga developers, publishers and retailers featuring the newest Amiga software and hardware at the lowest prices. The place where more Amiga companies introduce new products. An exposition of Amiga technology second to none in the world!

## Seminars

Each day will feature panels filled with Amiga information, supplied by working Amiga professionals in areas such as Paint Systems and Graphics, Professional Desktop Video and Amiga Expansion.

## Classes

Special intensive classes offering an Amiga education. Sessions are limited in attendance and taught according on a project basis. Course titles include Desktop Video, Programming, Music and Marketing. Call for schedule and availability.

***NOW PRE-REGISTER BY PHONE***  
***Call 800-32-AMIGA***

**IN NEW YORK STATE CALL 212-867-4663**

☐ Yes, I want to come to AmiEXPO - California

One day - \$15 \_\_\_\_\_

Two days - \$20 \_\_\_\_\_

Three days - \$25 \_\_\_\_\_

\_\_\_ Friday \_\_\_ Saturday \_\_\_ Sunday

Make Check of Money Order Payable to:

**AmiEXPO**

**211 East 43rd Street, Suite 301**

**New York, NY 10017**

NAME \_\_\_\_\_

COMPANY \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

For \_\_\_ MasterCard or \_\_\_ VISA Payment

Expiration Date \_\_\_\_\_

Account Number \_\_\_\_\_

Name as it appears on card: \_\_\_\_\_

Signature \_\_\_\_\_



# Shakespeare

## Color desktop publishing for the Amiga

by Barney Schwartz

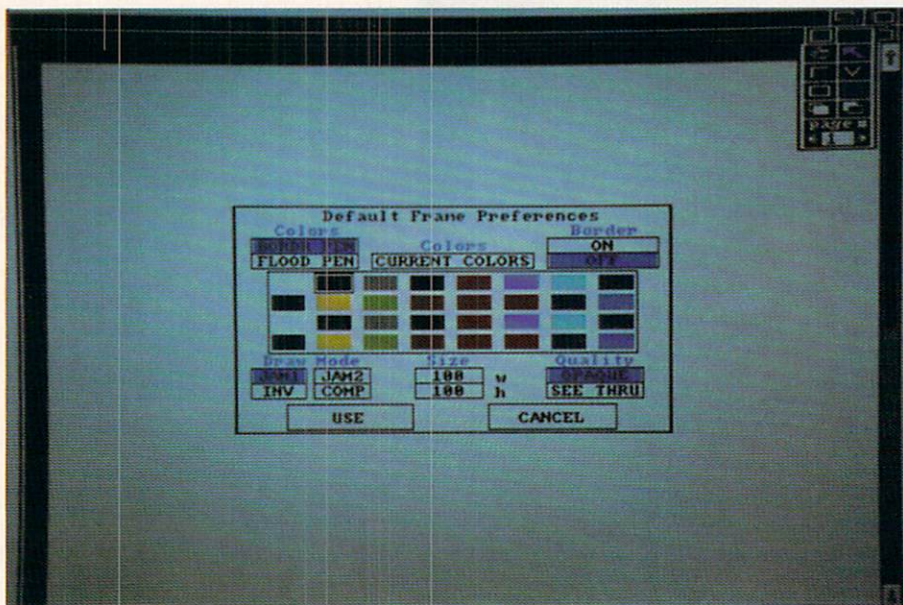
Shakespeare version 1.1 has finally been shipped to all registered owners of the original package. The upgrade expands on version 1.0, fixes numerous bugs, explains many puzzles and turns what was the first real color desktop publishing package for the Amiga into the second real color desktop publishing package for the Amiga.

Infinity Software was first to market with a color desktop publisher for Amiga when they distributed Shakespeare in March. Shakespeare was the only desktop publisher which allowed color graphics import, complied with the COLORTXT standard, allowed import of almost all fonts, and produced both postscript and readable dot-

matrix printouts. A lot has happened in the market since the introduction of Shakespeare. However, Shakespeare version 1.1 is still only the second desktop publishing package for Amiga that provides all those capabilities.

I won't attempt to compare version 1.1 and its improvements to version 1.0. Rather, I'll explore version 1.1 as if it were a new product (since it actually is).

Shakespeare is a page format and manipulation engine. It allows you to import, resize, and position, pictures, and/or text files. Shakespeare also allows to mix, match, shuffle, sort, and view your composed pages. When satisfied with your masterpiece, you can print on any preferences-supported printer or send your document to a postscript device.



*Shakespeare allows color graphics import and complies with the COLORTXT standard.*

### The Program

Shakespeare ships on two disks. The program disk is written on AmigaDOS 1.2, but the printer drivers are GAMMA 18 release. According to Infinity, "this means that at the time of shipment they are not in general distribution, and as such, are necessary for Shakespeare to print." The program disk includes full support for PostScript printers and fonts for use with PostScript only.

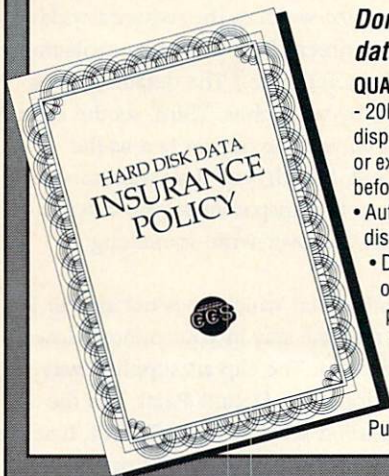
The second disk contains clip art and a :FONT directory of custom-built, clean fonts which look much better than those used by NOTEPAD. An obvious level of cooperation between INFINITY Software and the people at Interactive Softworks shows

(continued on page 12)



# QUARTERBACK™

NEW  
for the  
AMIGA



**Don't fumble around with your Amiga files. Let *QUARTERBACK* manage your valuable data. The Quarterback sneak scores every time!**

**QUARTERBACK** is a **FAST** Hard Disk to Floppy Backup Utility for the Commodore Amiga, featuring:

- Fast backup - 20MB in less than 40 minutes
- Uses two floppy drives for backup with automatic switching
- Builds, sorts, and displays catalog of files and subdirectories
- Provides Full/Subdirectory/Individual file backup/restore
- Includes or excludes files by name (with wild cards), file date, or archive bit
- Calculates the number of floppies you'll need before you start
- Handles files of unlimited length, unlimited subdirectories and unlimited files per subdirectory
- Automatically formats diskettes with no delay as it writes
- Sequentially numbers and date/time stamps backup diskettes
- Checks the sequence number and date/time stamp of each diskette before restoring files from it
- Detects bad disks during backup or restore
- Restores original date/time stamp, file notes, and protection bits on both files and subdirectories
- Runs from Workbench or CLI
- Produces backup/restore report to disk or printer
- Beeps for floppy change
- Accepts CLI parameters and batch command files
- Convenient/user friendly error recovery
- Multi-tasking
- No copy protection
- Works with all AmigaDOS compatible hard disk drives.

**You'll have fewer "time-outs" with *QUARTERBACK* managing your file backups.**

Put **Quarterback** on your team for only \$69.95 plus \$3.00 for shipping and handling, CA residents add 6% sales tax.



## Convert C64/C128 Files to the Amiga!

**DISK-2-DISK™** makes it easy and convenient to transfer C64/C128 files to and from the Amiga! **DISK-2-DISK** programs the Amiga model 1020 external 5.25" disk drive to read and write 1541/4040 and 1570/1571 disk formats including 1541 "flippies".

- Converts Commodore/PET ASCII to AmigaDOS standard ASCII and vice versa
- Transfers word processing text files (such as PaperClip, SpeedScript and Pocket Writer) to and from the Amiga for use with popular Amiga word processors
- Includes 3 public domain programs for converting C64 Koala, PrintShop and Doodle files to IFF format
- Finds and flags dialect differences between Commodore Basic and Amiga Basic files
- Provides **VALIDATE BAM** and **CHECK DISK** utilities (**VALIDATE BAM** verifies the directory structure of the 1541/1571 diskette; **CHECK DISK** reads every block of a 1541/1571 diskette to detect diskette errors).

*DISK-2-DISK requires the Amiga model 1020 5.25" disk drive.*

**Only \$49.95**

plus \$3.00 shipping and handling  
CA residents add 6% sales tax.



## Read/Write MS-DOS and Atari ST Disks on your Amiga

**DOS-2-DOS™** Transfers MS-DOS and Atari ST Files To and From AmigaDOS!

- Supports single and double sided 5.25" as well as 3.5" 720KB MS-DOS diskettes
- Reads/Writes 3.5" Atari ST diskettes (GEM format)
- Converts ASCII file line-ending characters and provides Wordstar compatibility
- Supports full directory path names, with wild cards in the file names
- Allows selection of MS-DOS and AmigaDOS subdirectory and displays sorted directory listing
- Formats 3.5" and 5.25" MS-DOS diskettes
- Provides duplicate file name detection with query/replace options
- Provides **TYPE** and **DELETE** commands
- Permits renaming of files where file name restrictions occur
- Remains resident to permit AmigaDOS disk swapping.

**Only \$55.00**

plus \$3.00 shipping and handling  
CA residents add 6% sales tax.



# Central Coast Software™

268 Bowie Drive, Los Osos, CA 93402 • Telephone (805) 528-4906 • FAX (805) 541-4745

Dealer Inquires Welcome

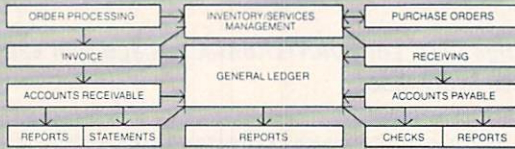




## Trade the Rest for the B.E.S.T.!

One program powerful enough to handle all of your basic business needs! Designed for people who are **NOT** accountants or computer experts.

"A TRULY INTEGRATED ACCOUNTING SYSTEM"



— LIMITED TIME OFFER —

**Trade in ANY  
business accounting program  
for \$50 credit on B.E.S.T.  
Business Management!**

B.E.S.T. Business Management is available for the Commodore Amiga, Atari ST, and MS-DOS.

Offer good only at participating dealers!  
Offer expires October 31, 1988

— MONEY BACK GUARANTEE —

**Call for more information!**

**Call 1-800-289-2544 for the dealer nearest you!**

*(Shakespeare, continued from page 10)*

through. The fonts included on Shakespeare closely resemble those available from Interactive. These fonts are for use with preferences printers and yield excellent results on my Epson JX-80.

Shakespeare uses all four display modes: hi-res (640 x 400), medium-res (640 x 200), interlace (320 x 400), and low-res (320 x 200). The maximum number of colors allowed in each mode are the standard 32 low-res and 16 hi-res. Switching display modes allows you to view different sections of your document, but never the entire page. To see the entire page, you must select DISPLAY from the WINDOW menu. You may limit the colors used to conserve chip memory. This is a good practice, since chip memory is always in short supply with graphics intensive programs. In fact, if you know you will be printing black and white or gray scale only, you may want to use just two colors.

### Running the Program

Before invoking the program, you must set your printer with preferences. Set the printer you intend to use and the density you desire. Make sure you have selected the correct port (parallel or serial) and switched to color or black and white. If you want a specific dithering mode, select it, too. The other choices within preferences can be ignored—they don't seem to have any effect on output. The technical support team at Infinity Software informed me that users only need to set the correct printer, as all other choices are controlled within the program.

Once you have set your printer, double click on the Shakespeare icon and you are ready to start composing. The program defaults to hi-res, sixteen colors, inches as your unit of measurement, and rulers and borders on. These parameters can be changed before calling the program by clicking the program icon

and selecting INFO. From within INFO enter any required changes in the tool types line in CAPS. This feature cannot be accessed through the CLI.

You should do a few things before putting your first word or graphic on the page. First, set the color palette to the colors you prefer. Second, call Page Setup from the Project menu and set the page size width to the required width of your printer. (This corrects a problem with preferences.) The default page is eight by ten inches. Third, set the default font so you can import text as the default. Fourth, you may want to set all frames to transparent so you can see when you over-write something.

Another hint which does not appear in the manual: stay in your printer's lowest resolution. The clip art supplied was developed on Deluxe Paint, and the resolution is set to 75 by 75 DPI. It seems as though a lot of what the program does is centered around 75 by 75 DPI. You may need to remember this if you import graphics from sources other than paint programs.

The manual says you should use "Print to Fit" to get proper sizing, and that you should not adjust the DPI number unless you own a "non-preferences" supported printer. However, Infinity tech support will tell you to use 1-to-1 printing unless it appears ghosted, in which case you're stuck with "Print to Fit." Experiment with the different print option until you discover which ones produce the best results.

The manual suggests: "Although not required, it might behoove you to put all text and graphics files on a data disk BEFORE you start to build your issue." This maneuver greatly reduces disk swapping and may also prevent a GURU visit. There are enough of these unpleasant incidents, especially when you resize off the screen or attempt to print with insufficient memory. Version 1.1 corrected past problems with memory management.



Now that we have completed the preliminaries, it is time to create a document. Boot Shakespeare and open or create an issue. Next select the Page menu and Create Columns, and make your desired selections. To import text, move to the Window Menu and select Chart. A file requester appears allowing you to determine the file you wish to import. Highlight the desired file and load it. Using the Place tool, place the text in each open frame until you reach the end of your article. A similar procedure is used to put graphics on the page.

First, call the Chart Menu and select the desired file. Use the Place tool to put your graphic in the issue. Notice, you did not open a text box. To insert graphics, you simply select a file and place it. Once this is done, use the Finger gadget in the ToolBox to move the graphic.

Memory Cleanup is an added selection in the BOING (\*) menu. This option forces any moveable part of the document into fast RAM (if available). The selection works well to free memory. In the About Shakespeare billboard is a memory use indicator. Monitor this frequently throughout the production cycle. This way you can forestall a nasty GURU due to insufficient memory. Shakespeare can still allow an attempt to print, even if memory is too low to allow formatting of a document. Be warned!! If this occurs, you lose all unsaved data. Save your issue frequently.

#### About the Manual

The 191-page spiral bound manual is one of the best pieces of software documentation I've seen. The foreword explains the manual and operating on the Amiga. A quick reference to the toolbox and keyboard equivalents appears before the Table of Contents. Getting Started explains terms and gadgets to the uninitiated. Chapters two, four, and five are comprehensive tutorials, while chapter three explains layout procedures. Chapters seven, eight and nine are Technical Reference, Hints and Tips, and Design Considerations.

Chapter six deals with printing. An appendix deals with PostScript and includes an index.

If you progress from the beginning to chapter five, you get a good feel for the program. The tutorials do an excellent job of covering all screen aspects of the program. When you have completed them (I highly recommends that you complete them all.), you can do everything except print to a dot-matrix printer.

In chapter six, "Preferences & PostScript Printing," the manual falls short. The addendum shipped with version 1.1 goes a long way to clarify the information in this chapter. Three pages are packed full with information about preferences printers, scaling, font sizes, page sizes, using more than one printer on your system, color printers, high resolution and black-and-white printing. Another two pages are dedicated to explanations of the new printer drivers included from

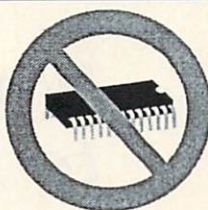
CBM. This information details each print driver's capabilities and best methods for use. An in-depth explanation of the new preferences screens and their use is also included.

Unless I am mistaken, the only presently available means of achieving color output is from dot-matrix or overlays. Particular attention should have been given to this portion of the manual.

The manual has a good explanation of Postscript operation with reference to the appendix for further Postscript help. But Shakespeare is a color desktop publishing package, and postscript does not print color.

#### Overall Impression

Shakespeare version 1.1 is an excellent program that does something not all programs do—it does what it says it will. The screen control and format operations



*We cannot lower the price of memory chips, but we can LOWER the price of the INSIDER so you can AFFORD the high price of chips.*

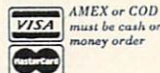
## MEMORY CHIP BUSTERS

### The Original INSIDER

0K One Meg Board with clock  
Fully socketed ... Just plug in the\* chips  
for the A1000 only  
90 Day Warranty



**\$150.00** plus \$3.00 Shipping



AMEX or COD  
must be cash or  
money order

NO Dealers or Distributors - Prices too LOW!  
\*Requires 32 256K chips @ 150 or 120 ns  
Order direct from the manufacturer only:



#### Michigan Software

43345 Grand River  
Novi, MI 48050  
(313) 348-4477  
Mon through Fri 10 to 5

(continued)



really shine. Shakespeare, the page integrator, is truly the first and second color desktop publishing program available for the Amiga.

Shakespeare is the best of the current crop, and its features may be enough to lure PC users to this wonderful machine. It is better than many of the existing "mono-tone" publishing packages on the market. Mouse control is very effective, so I find no fault with the absence of keyboard equivalents for some menus. In fact, leaving the mouse would slow you down in this graphics intensive environment.

Some improvements are still needed, but judging from this edition, they will be provided. Slight scaling improvements to allow for higher density printers would be nice. A built-in text and graphics editor would be really helpful, but may be out of scope. An automatic low memory warning is a must for this graphics intensive package to become a field leader. And some way of resetting the issue to your last action if memory is unavailable would be better than GURUing out.

Overall, if Infinity follows through with further improvements and program expansion, they've got a sure winner in Shakespeare. Buy Shakespeare if you do not need color overlays or black-and-white laser prints, and you want color. If you want some other capabilities, they're just over the horizon.

(To all who anxiously await information about AmigaTeX, I apologize. I missed my deadline, but will present an AmigaTeX article soon.)

Send your questions about Amiga desktop publishing to:

**Barney Schwartz**  
c/o Amazing Computing  
P.O. Box 869  
Fall River, MA 02722

## Would you like to see a menu?

Shakespeare is designed to run from the Workbench and uses Intuition very effectively. The menu bar contains the following menus:

### 1. Boing Menu (\*)

- a. About Shakespeare
- b. Help
- c. >> Change Palette
- d. Change Resolution
- e. Memory Cleanup

### 2. Project Menu

- a. New Issue
- b. >> Open
- c. Save Issue
- d. Page Setup
- e. Print
- f. QUIT

### 3. Edit Menu

- a. Cut
- b. Copy
- c. Paste
- d. Delete
- e. Copy Text Prefs
- f. Paste Text Prefs

### 4. Style Menu

- a. Plain
- b. Bold
- c. Italics
- d. Underline
- e. FONT...

### 5. Page Menu

- a. New Frame
- b. Copy Frame
- c. Create Columns

### 6. Window Menu

- a. Edit
- b. Tool Box
- c. Display
- d. Chart...
- e. Text Prefs...
- f. Frame Prefs...

There is also a tool box that resides in its own window. This tool box can be repositioned, removed, or paged behind the active window. Gadgets on the tool box are:

- 1. Close Button
- 2. Drag Bar
- 3. Graphics move (hand with pointing finger).
- 4. Arrow Tool. Standard Amiga arrow pointer.
- 5. Frame Tool. Create and delete frames.
- 6. Wedge Tool. Allows insert a text frame.
- 7. Border tool. Draws borders around text or graphics.
- 8. Edit Frame. Draws dotted line around frames.
- 9. Frame order tools. Moves frame like FRONT/Back gadget.

Most menu selections have keyboard equivalents. These are listed on page vi of the manual. All but five of them use right Amiga plus a letter. The five loners are F1, F2, F3, F10, and HELP. F1 through F3 are used to control motion of windows while moving a window. I found that the only key I tend to use is F10 (refresh screen). All other screen manipulations are accomplished efficiently under mouse control.

—Barney Schwartz





# The Kideo Tapes

by John Dandurand

When first confronted by a blank DPaint screen and a waiting mouse, most of the kids just froze up or sort of tentatively wiggled the mouse. The prevailing mood was one of impending disaster. But by the end of the class, we had created a space monster that vaguely resembled a blobby green kitchen table. Our alien landscape looked like grey smog on a pepper-spattered, mustard-yellow background. The students, sixteen fourth graders at Lanier Elementary School in Gainesville, Georgia, were elated at the prospect of producing their own Kideo, despite their timidity when faced with 6400 empty pixels. (Who isn't intimidated first time around?) Nevertheless, we now had the beginnings of our first animation.

I entered the Georgia Arts-in-Education program to teach kids how to build clay drums, write their own lyrics to songs, sing together (sometimes), and so forth. My years on the road as a musician provided scant preparation for this plunge into elementary education. With a daughter ready to enter school, though, I was spurred to become involved in the school arts scene. (Marrika, now 7 years old, frequently appears with me for children's concerts as a violinist and singer.) None of the teachers or students seemed to know quite what to make of me at first which helped even things out.

My first long-term residency, at Len Lastinger Elementary in Tifton, GA, focused strictly on music. At Len Lastinger, I encountered the Apple IIe and fell in love with computers. At first I loved the "magic typewriter" of word processing, next because I thought I had discovered the ultimate machine: the "do-everything box." I soon found that the Apple II had some severe limitations as an artistic tool.

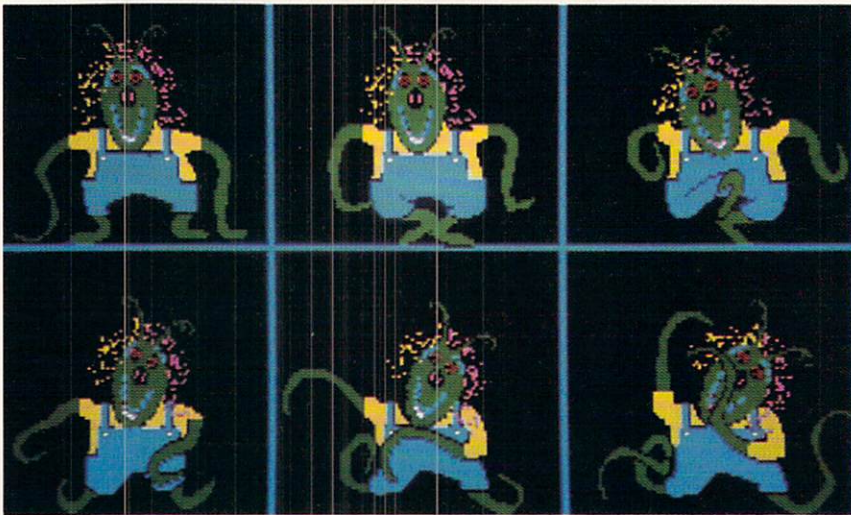
Prior to the Len Lastinger residency, I was a computerphobe, but my familiarity with sound reinforcement and recording equipment helped ease the transition. Regrettably, many teachers—despite daily access to computers (Apples, for the most part)—still



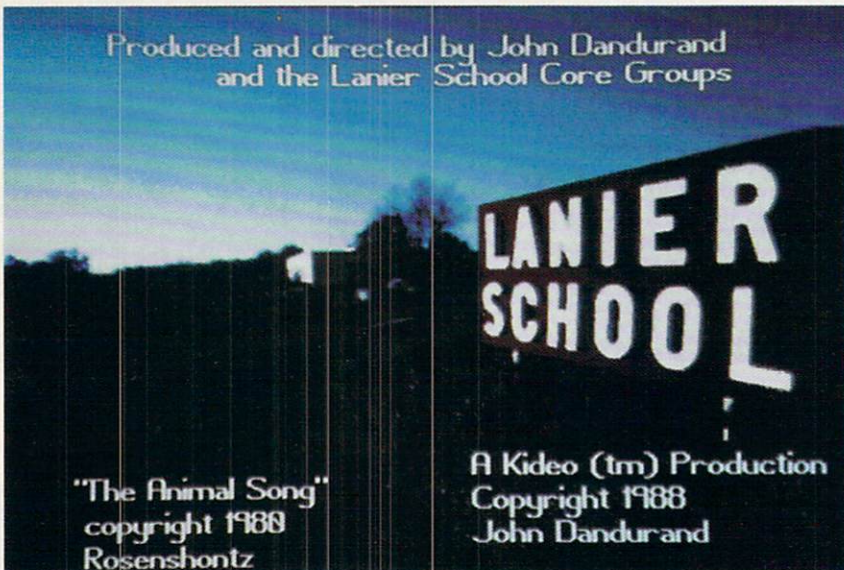
Sample of Kideo scene created in DeluxePaint.

(continued)





"Monster Dance" created using DeluxePaint.



can't or won't incorporate them into classes for more than the minimum time required by the school. With access to a more user-friendly machine like the Amiga, some of these folks might come out of their shells.

While visiting fellow resident artist Bond Anderson (flutist, synthesist, instrument builder and C64 aficionado) at his home in Parrott, Georgia, I ran into an A1000 in the hands of Bond's father-in-law, a jolly old elf of a retired illustrator named Lew Tilley. Lew showed me around DPaint, let me copy a few wallet photos onto

disk with Digi-View, and introduced me to Marble Madness. I left the Tilley-Anderson complex—a rambling series of interconnected concrete block octagons of varying heights, filled to overflowing with books, musical instruments, VCR's, CRT's, and no less than eight computers, mostly Commodores—a changed man.

My residency at Lanier Elementary was coming up. Fired with the zeal of a total convert (I got into arguments with good friends over computers in general, and the Amiga in particular.), I resolved to do a series of classes combining computer

music, graphics, and animation with singing and live video. I wrote Frank Leonardi, then Commodore's National Sales Manager, with a proposal outlining the project.

To my good fortune, Commodore agreed to send me an A1000 system (512K, 2 drives, RGB monitor) for the project. Aegis and Electronic Arts donated software to get us started. A surprising number of corporations have since chipped in. I could not have even begun this project without their help. The level of artistic and technical sophistication the ongoing Kideo program enjoys is entirely due to this type of continued support.

Once I received the machine and set it up, I had to figure out how to use it. Commodore bundled DOS 1.1 with my machine, and it produced error messages every time I turned around. Despite that difficulty, I learned how to manipulate DPaint and Animator with alacrity. I had considerable incentive to do so, since I was committed to a project that I'd never attempted before—using unfamiliar equipment—to be completed in 10 weeks! All this, coupled with a new baby and a twice-weekly 350 mile commute from where I was living in Tifton, GA to Gainesville, GA where the residency took place, helped raise my blood pressure. (Especially when I'd stayed up late to work, only to have the Guru appear and trash my foolishly unsaved files.)

The one package I didn't have the time to figure out until after the first project was complete was Videoscape 3D. My inexperience and the complexity of the object editor interface both contributed to this.

After that first class, our Kitchen Table Creep evolved fairly rapidly into a full-fledged Bug-Eyed Monster, complete with waving tentacles and a pair of Farmer Johns. The background emerged from the smog sporting a large lake, and the kids created a flying saucer, piloted by a second Bug-Eyed Monster. Its job was to beam Kitchen Table up and down and to zoom around the screen. We animated Kitchen Table by creating a series of small frames with DPaint and



importing them into DVideo. DVideo allows you to "page-flip" the frames while moving the frame box around the screen over an IFF background. We used this same technique to make a monkey eat a banana.

Titles were created with DVideo's text line feature, along with the aforementioned fonts for a scrolling effect, the custom "polygon text" feature for rotating titles, and color cycling. During the title sequence, one of the kids acted as narrator, reciting the words and adding humorous interjections.

While it is a wonderfully complex and flexible program with innumerable features, DVideo uses only 16 (8 foreground and 8 background) of the famous 4096 colors. You can avoid this to some extent by switching palettes.

Aegis' Animator, which we also used, is a dedicated animation program. What the program lacks in extras like music or slideshow modules, it more than makes up for in its ability to produce quality 2-D animations. We used Animator to do an animation of a goony bird metamorphosing into a pink and purple whale. We used the "morph" feature to change the shape of our creature during the course of the script.

The effect was very different from the traditional frame-by-frame animation we used in DVideo. While the user defines where the morph begins and ends (bird and finished whale), the program decides what the morph looks like in between. Aegis labels the whole esoteric process with the equally esoteric name: "polymorphic tweening." This same kind of tweening is used to interpolate a series of sequential frames in 3-D "polygon-mesh" animation programs. Results can be similarly unpredictable if you're not careful.

According to the guidelines set down by the National Endowment for the Arts, the artist is expected to work intensively with "core groups" of children who are selected for their aptitude in that particular art form. For the "Wild Wishes" project at Lanier, I worked with four core

groups—two for music and two for graphics. I started with roughly sixty kids in all: thirty fourth graders to create graphics and thirty third graders for the soundtrack.

Although the students are supposedly chosen for their abilities, in a school that has had several artists, the core groups are limited to kids who haven't participated before. So some of the singers couldn't exactly sing, and many of the kids waiting to take their turn at the Amiga just didn't have the patience. When we narrowed it down, the computer groups had seven and eight each, and I consolidated all the singers into a single group of twenty.

The size of the computer groups directly depends on the number of computers available. No more than two students should have to share the same workstation. One student to one machine is the ideal ratio to maximize class time use and teacher effectiveness. I had one Amiga. (A possible way around this numbers game in the future educational environment might be an Amiga that is multi-user, as well as multi-tasking.)

I can hear some parents out there complaining about how unfair this kind of weeding out is. All I can say is, if you want your kids to be exposed to the arts, get more art, artists, Amigas, and most of all, more involved in your kids' school.

During this residency, I helped all my kids (over 400 from the first through the fourth grade) construct over fifty hand-shaped clay drums with real rawhide heads. (Try that with your first-grader sometime!) I also taught them a variety of songs from America's musical past—performed with the instruments on which the songs were originally played.

For the musical end of our Kideo the kids chose a goofy tune called the "Animal Song" (Copyright Rosenschon, 1980). We made up new verses and left the chorus intact. My original concept was to create a harmony part for one group, have the second group sing the melody, while a third group beat out the rhythm on their newly made clay drums.

Using a DMusic score, the Amiga would provide the "bed," or background tracks, while I strummed accompaniment on my 1909 Gibson Mandocello. (A mandocello is a guitar-sized mandolin, tuned like a violoncello, used in the mandolin orchestras that were popular around the turn of the century.) By listening to different harmony lines and comparing them one by one, my music core helped me select the lines we ended up using in our final score. The results would then be recorded on the audio track of the video we planned to make when we had the thing completely rehearsed.

Well, things didn't work out quite the way I planned. With only one notable exception, after four weeks of rehearsal, my drummers showed a lot more aptitude for producing random cannonades than any kind of a usable, steady beat. Consequently, although the drummers are visible in the finished tape and their hands move, there are no audible drum sounds! After reviewing the takes, we found the drums so off beat that they were distracting. I re-recorded the entire audio track—once again synchronized to the DMusic arrangement—with my own 4-track studio gear with all the same parts, except for drums.

The tempo of the DMusic bed was exactly the same both times, so when Louis Bailey (the videographer) and I edited the takes and mixed them with the credits and animations, we were able to synchronize the singing on the audio track with the movement of the kids' lips on the video—a neat trick, but one that requires professional insert editing equipment (accurate to two frames). ACTS-Network TV-17 generously gave us access to their editing facilities in Tifton, Georgia with Sony 3/4" source, edit and frame control equipment. If you try this, bear in mind that synchronization is only precise for a maximum of twenty or thirty seconds because the tape itself stretches slightly during the record/edit process.

The easiest task was constructing the credits screens. All the kids brought in personal photos, and we used Digi-View

*(continued)*



## CBTREE, the easiest to use, most flexible B+tree file manager for fast and reliable record access ON SALE NOW.

### **CBTREE... Includes over 8,000 lines of 'C' source code FREE!**

Since 1984, thousands of 'C' programmers have benefited from using CBTREE.

**Save programming time and effort.** You can develop your applications quickly and easily since CBTREE's interface is so simple. You'll cut weeks off your development time. Use part or all of our complete sample programs to get your applications going FAST!

**Portable 'C' code.** The 'C' source code can be compiled with all popular C compilers for the IBM PC including Microsoft C, Quick C, Turbo C, Lattice C, Aztec C and others. Also works under Unix, Xenix, AmigaDos, Ultrix, VAX/CMS, and others. CBTREE includes record locking calls for **multi-user and network applications**. The CBTREE object module is only 22K and is easy to link into your programs. You don't even pay runtime fees or royalties on your CBTREE applications!

**Reduce costs with system design flexibility.** CBTREE allows unrestricted and unlimited designs. Reduce your development costs. You define your keys the way you want. Supports any number of keys, variable key lengths, concatenated keys, variable length data records, and data record size. Includes crash recovery utilities and more.

**Use the most efficient search techniques.** CBTREE is a full function implementation of the industry standard B+tree access method, providing the fastest keyed file access performance.

#### **Database Calls:**

- |                          |                              |                          |
|--------------------------|------------------------------|--------------------------|
| • Get first              | • Get greater than           | • Insert key             |
| • Get last               | • Get greater than or equal  | • Insert key and record  |
| • Get previous           | • Get sequential block       | • Delete key             |
| • Get next               | • Get partial key match      | • Delete key and record  |
| • Get less than          | • Get all partial matches    | • Change record location |
| • Get less than or equal | • Get all keys and locations |                          |

### **CBTREE is only \$159 plus shipping - a money-saving price!**

We provide **free telephone support** and an **unconditional 90-day money back guarantee!**

To order or for additional information on any of our products, call **TOLL FREE 1-800-346-8038** or **(703) 847-1743** or write using the address below.

**NEW CBTREE add-on products to make your programming easier. AVAILABLE NOW!** • Adhoc query, reporting system; • SQL interpreter.

If you program in 'C', sooner or later you're going to need a B+tree. Don't delay until you're in a crunch, plan ahead, place your order for CBTREE now. Orders shipped within 24 hours!



**PEACOCK SYSTEMS, INC.**  
**2108-C GALLOWS ROAD**  
**VIENNA, VA 22180**

to store them to disk. With DPaint's text insertion feature along with Amiga and Zuma fonts, we named the students and staff involved and thanked our donors. I

supplemented the kids' pics with my own 35 mm slides of the school and nearby environs.

To get best results with Digi-View, I used a light box with 5000K bulb, masked everything but the slide (this method works *only* with slides), and focused down very close to the slide by moving the copy stand as low as it would go without touching the slide. I coarse-focused with the "mech" adjustment on my Panasonic WV1410 CCTV camera (which actually moves the tube within the camera), and then fine-focused at the lens. Other light sources in the room must be turned off during this operation. The resulting lighting is always very even with well-balanced colors and no reflections.

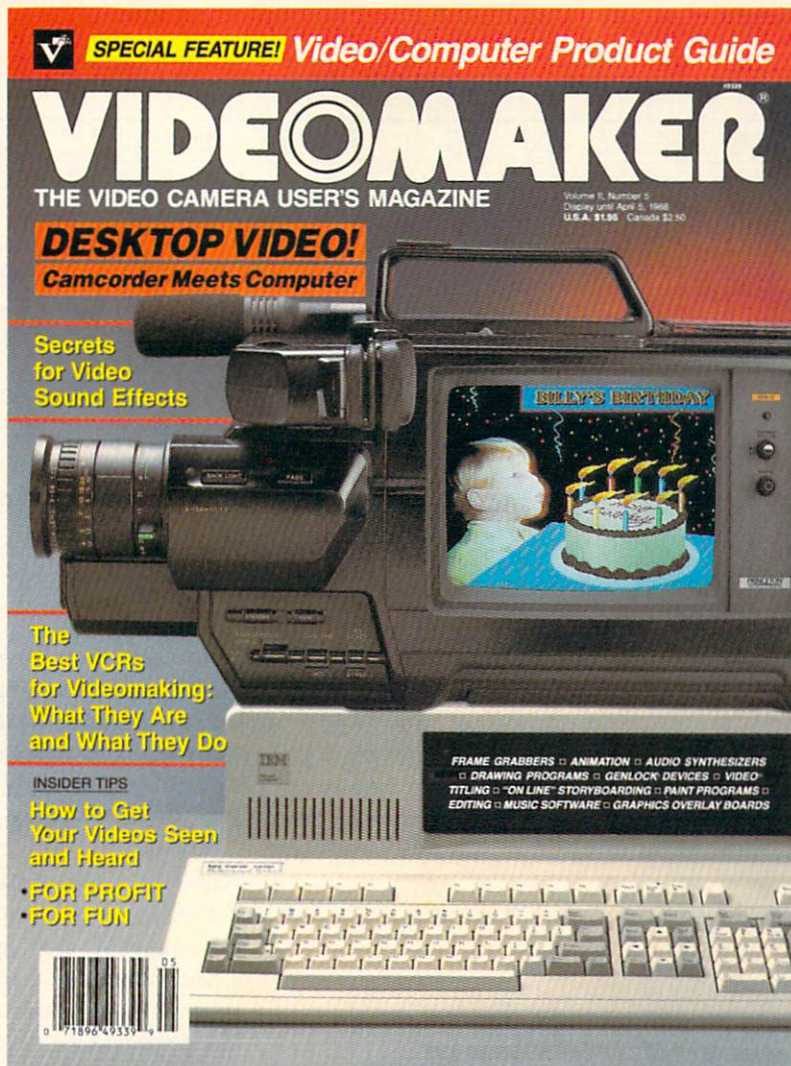
Copying the prints the kids brought in was more difficult (many were unusable due to under- or over-exposure), but we improved image quality by manipulating the images with Digi-View's excellent color controls before storing them. In one case, we took pictures using outdoor film indoors (by accident), then fixed the color balance after digitizing!

During these sessions, my role was as an interface between the kids and the machine, interpreting their artistic ideas with specific program commands. The hardest thing for me to do was to just turn them loose, but once they had some familiarity with DPaint, I found their pixelized paintings very expressive and well-suited to the medium. Their colors tended toward garish combos, but subtlety is not the strong suit of an eight-year-old.

All this commuting, computing, harmonizing, and editing came to an all-too-brief halt in December when my Lanier residency was over. Finished tapes were sent to the school, the Georgia Council for the Arts, Commodore, and others who supported our efforts. It has since been shown on WHNT-TV19, a CBS affiliate out of Huntsville, Alabama, and statewide on Georgia public TV.

•AC•





*What do you get when you cross  
an Amiga with a video signal?*

**Creative potential beyond belief.**

*What do you get when you cross  
creative potential beyond belief with ongoing,  
comprehensive coverage of the phenomenon?*

**Videomaker magazine.**

Everyone's getting into the act: Computer buffs are focusing on video. Videomakers are plugging into computers. And computer/video enthusiasts at large are tuning in to *Videomaker* magazine for tools, tips, and techniques galore—the stuff computer/video magic is made of. "The Video Camera User's Magazine" also happens to be the video/computer user's magazine. Every issue.

**Plug into the possibilities . . . with Videomaker.**

Focus on the creative, practical tips you'll find in every issue and zoom in on our **LOW CHARTER SUBSCRIPTION RATE!** Save 33% off the regular subscription rate that future subscribers will pay.

☐ **Yes!** I want to subscribe for one year (6 issues) at \$9.97.  
☐ Payment enclosed ☐ Bill me

Name

Address

City  State  Zip

S89C

**VIDEOMAKER®**

P.O. Box 3727, Escondido, CA 92025



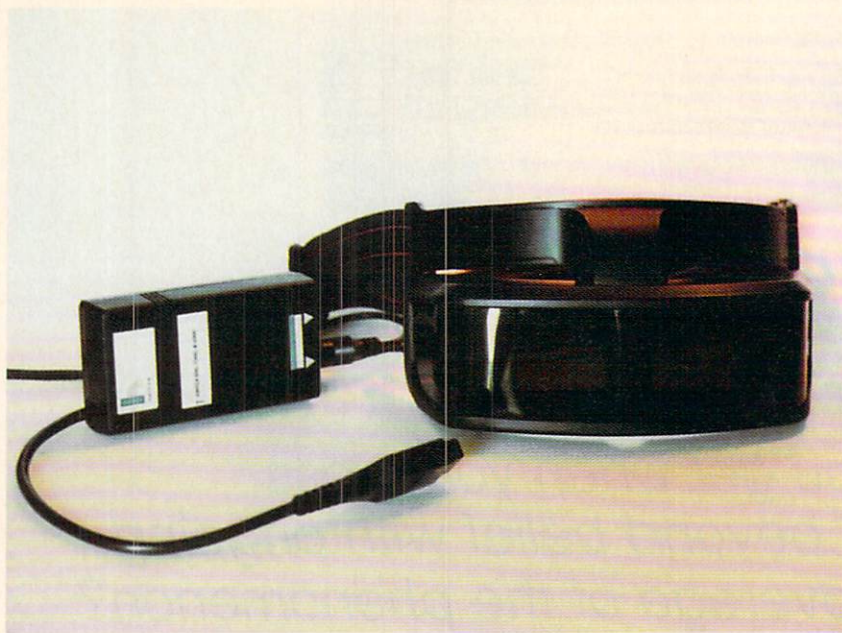
# X - S P E C S 3 D

by Steve Hull

GENie: LightRaider

People Link: St.Ephen

THREE DIMENSIONAL GRAPHICS! is probably the most overused advertising phrase in entertainment software today. It seems as if every title from sports to flight simulators claims the kind of realism which results from simulating depth on a screen that has only height and width. Most of these titles resort to pretty simple methods of delivering on the promise—light and shadow, perspective, or the kind of multiplane animation first perfected for cartoons by Disney. Effective? By and large. Three dimensional? No.



Haitex' X-Specs 3D use shuttered LCDs synchronized with left and right interlaced images to produce a striking 3D effect. The interface module (left) plugs into the #2 joystick port and will accommodate two pairs of glasses.

Haitex Resources of Carrollton, Texas, has come out with a product that could redefine the meaning of 3D in Amiga graphics. X-Specs 3D is a hardware device consisting of a pair of shuttered-LCD glasses and a compact interface that plugs into the #2 joystick port. Coupled with the right software, X-Specs produce an illusion of depth so convincing you might sprain a finger trying to reach through the screen!

## 3D OR NOT 3D

To understand how X-Specs achieves this effect, you must first understand the principles behind natural stereoscopic sight. A simple experiment will help.

Stretch your arm out in front of you and point your index finger up. Look straight ahead, then view your finger using only your right eye.

Switch to a left-eye view. Alternate back and forth a couple of times, and you'll notice that each eye sees a different picture; relationships between objects at varying distances change noticeably. Your mind's merging of these two images allows you to perceive depth; with one eye closed, even the real world appears in only two dimensions.

Conventional video images provide only a one-eyed view. To achieve the effect of true 3D, the left and right eye must see the image from slightly shifted perspectives. This has been done a number of ways.

Three-dimensional pictures gained wide popularity before the turn of the century. Two cameras were placed side by side, and triggered simultaneously. These left and right photos were placed in hand-held stereo viewers. This is the same principle behind the Viewmaster stereo slide viewers.



The comics flirted briefly with 3D by using glasses with one red lens and one blue lens. The left and right images were printed in red and blue, the tinted lenses masking out the right image from the left eye and vice versa. Infocom's *Leather Goddesses of Phobos* includes a 3D comic which uses this method, as does a 3D Breakout game for the Amiga available in the public domain. While it offers true 3D, this method's greatest disadvantage is the limited range of color available.

Hollywood produced a spate of movies in the 50s and 60s that used polarizing glasses to produce 3D effects. Like the red/blue glasses, the polarizing glasses were clumsy and uncomfortable for eyeglass wearers. More significantly, the problems caused by the twin cameras' incorrect foreground/background spatial relationships virtually guaranteed headaches after two hours.

X-Specs uses a technique only recently made possible by liquid crystal display technology. Liquid crystal displays—LCD's—are most commonly found in digital watches. The LCD element, invisible when switched off, turns an inky black when energized. The lenses in X-Specs are actually high-speed LCD shutters, the left and right lenses blacking out alternately every 60th of a second. Matched with a video display that alternates left and right images at the same frequency, the X-Specs lenses allow the left eye to see only "left" images, and the right eye to see only "right" images.

#### THE RESULT: INCREDIBLE!

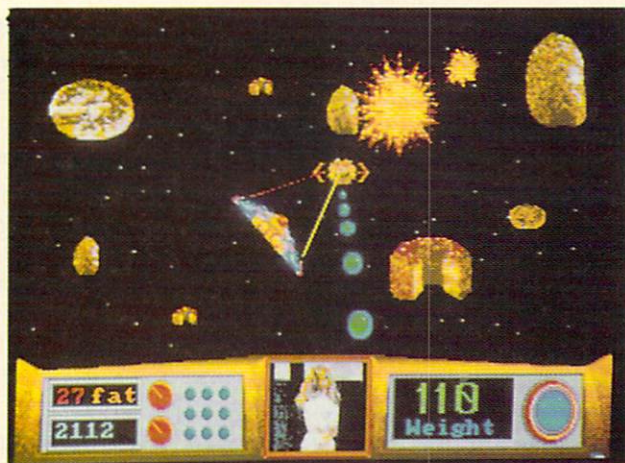
The results are pretty amazing. Included with the X-Specs glasses is a disk full of demonstration software: delicately rendered molecules turn slowly in real time; a digitized cat leaps out of the monitor; spinning wire cubes fly out of the screen. The most impressive demonstration of X-Specs' 3D capability comes in the form of a game: *Space Spuds* is a wacky outerspace adventure in which you pilot a ship through a never ending field of asteroid-sized potatoes and pastries, dodging and blasting lest an on-

screen character eat too many and explode! The game, written by Libyans in Space programmer John Schultz, features three dimensional effects that appear to extend cleanly a foot and a half behind the screen. The Amiga's crystal-clear graphics only enhance the illusion. Haitex representatives report mind-blowing results when playing *Space Spuds* on a wide screen TV!

#### EXAMINING THE HARDWARE

It is obvious by examining X-Specs that Haitex put a lot of thought into their design. The glasses are molded out of sturdy high-impact plastic, with a tinted wraparound cover that makes wearers look a little like the Terminator. The glasses rest not on the nose, but against the forehead, held in place by an elastic headstrap. A soft rubber gasket on the headpiece makes extended wear comfortable. There is also plenty of room to accommodate all but the largest eyeglass frames, a boon to prescription lens wearers, which alone is enough to set X-Specs apart from cheaper models manufactured for other computers.

The generous 7-foot connecting cord has stress relief built into three points in the glasses, and plugs into a small plastic interface which in turn plugs into the Amiga. The interface includes two jacks, allowing two sets of X-Specs to be used simultaneously.



*Space Spuds, a futuristic romp through a high-calorie asteroid field, is included in the X-Specs 3D package.*

#### DDD-PAINT

Besides the game and graphics demos, the X-Specs 3D package includes a program to display user-created 3D graphics screens. By using the swap screen feature of such programs as *Deluxe Paint* and *Photon Paint*, aspiring Amiga artists will easily be able to experiment with creating left images and right images which the 3D display program will then integrate. Video digitizers such as Newtek's *Digi-View* should present especially interesting possibilities.

*(continued)*



*G. Shawn Glisson "The Amiga showcases X-Specs 3D in a way that really gives it an unfair advantage over other systems," says Haitex President G. Shawn Glisson. "That's why we decided to introduce the glasses first on the Amiga."*



## DRAWBACKS

Though X-Specs 3D is a very good 3D system, it is not perfect. Its greatest limitation is the Amiga monitor, not the glasses. Because the Amiga monitor displays interlaced fields (not frames) at a rate of sixty per second, this limits the speed at which the LCD shutters may operate. This produces a very slight flicker, not noticeable in all applications. The only way around this would be to use a video output system and monitor with a faster refresh rate—say, 120 Hz. Monitors in this class start at about \$1,500. I can handle a little flicker.

The flicker becomes much more noticeable when the glasses are used in conjunction with fluorescent lights. Fluorescent lights flicker at 60 Hz, which sets up a "beating" which can be quite annoying. Best solution? Turn 'em off. I found the 3D effect most pleasing in a darkened room.

Software development for X-Specs or similar systems has some built-in challenges. Besides the geometry of matching spatial relationships correctly (getting it wrong gives viewers a headache), programs must be able to refresh the screen twice as fast as conventional screens (which refresh only 30 times per second). This can also increase the amount of memory required for one frame.

## YOU WANT MY OPINION?

Personally, I hope X-Specs 3D takes off like a rocket, and it just may. The list of software producers who have requested developer's kits reads like a *Who's Who* of Amiga publishing. Applications currently being explored range from true 3D drawing programs to intricate molecular modelling. "We find it interesting to have a product the physicists like as much as the kids [do]," remarks Haitex President G. Shawn Glisson.

Interesting indeed, though it should not be surprising to find quite a bit of...uh..."depth" to the X-Specs audience. At \$125, it won't be an impulse buy, but for those who must have the ultimate in Amiga graphics, it doesn't get any better than this.

•AC•

## X-SPECS 3D

Haitex Resources

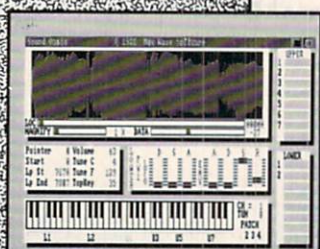
208 Carrollton Park Suite 1207

Carrollton, TX 75006

(214) 241-8030

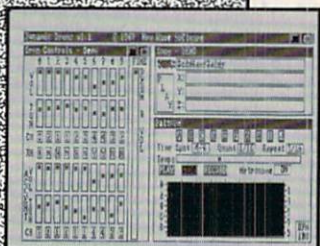
Suggested price \$125.00

The author wishes to thank Wade Bickel for his assistance in explaining 3D theory and technique.



**Sound  
oasis**

\$99<sup>95</sup>



**dynamic  
drums**

\$79<sup>95</sup>



**DYNAMIC  
STUDIO**

\$199<sup>95</sup>

The internal sound capabilities of the Amiga are better than that of any other personal computer. These capabilities mean nothing though, without quality digital sounds, which up till now have been scarce. Sound Oasis gives Amiga owners access to a large library of studio-tested digital samples, by using the Amiga's built in disk drive to read disks made for the Mirage Digital Sampling Keyboard. Sounds can then be played from a MIDI keyboard, the computer keyboard, or saved as an IFF standard file. Mirage is a trademark of Ensoniq Inc.

Transform your Amiga into a professional-quality drum machine with this software package. Easier to use than hardware-based drum machines because everything is displayed graphically on screen. Enter drum patterns quickly and easily in real time with visual feedback and editing. Create realistic drum tracks with any of the 100 drum and percussion samples that are included or use your own unique IFF one-shot samples. Dynamic Drums also has full MIDI implementation and even becomes velocity sensitive when triggered from a MIDI keyboard.

A powerful MIDI sequencer that takes full advantage of the Amiga's sound, graphics, and sophisticated user-interface. Dynamic Studio is perfect for professional applications due to its sophisticated editing capabilities and SMPTE support. It is also ideal for home studios, because in addition to sequencing MIDI instruments, Dynamic Studio has a built-in drum machine, and the ability to playback instruments translated with Sound Oasis.

**new wave  
SOFTWARE**

No Copy Protection

P.O. Box 438 St. Clair Shores, MI 48080 (313) 771-4465



# AmigaNotes

## A Look at 8SVX IFF Data Files

by Rick Rae  
76703,4253



In the past, this column has examined sound, discussed methods for synthesizing instruments, talked about the requirements of sampling, and so on. One thing we haven't done is dissect IFF storage of samples. Let's do that now.

### Binary Representation of Samples

Inside the Amiga, a sound waveform is represented by a string of eight bit bytes in twos complement notation. This perhaps deserves a bit of explanation. A lack of signal is represented by zero, or all bits off. Positive voltages are represented by numbers from 1 to 127, or 00000001 to 01111111 in binary.

Notice that only seven bits are used to represent these values. The eighth bit is a sign bit, which, when set, indicates a negative number. This means that values from 10000000 to 11111111 represent negative values ... but *what* values? This is where "twos complement" comes in. To convert a negative number to a positive number, begin by complementing all the bits. In other words, make all the zeros ones and vice versa. Thus the number 11111000 becomes 00000111. This is known as a ones complement. To perform a twos complement, add one for a final result of 00001000.

So—and this comes as a surprise to many people unfamiliar with binary—the value 255, the largest unsigned number that can be represented by eight bits, is -1 in twos complement notation.

Here is a string of bytes representing part of a sample in memory:

```
80 80 87 BF DF FF 17 1F 3F 7F 7F 78 50 20 F8 F1
```

Notice that this piece of waveform starts at 80 hexadecimal (10000000, or -128)—the most negative value possible. The values rise toward zero until at FF hex we reach -1; the next value is 17, which is positive. The waveform continues rising until it reaches 7F, the largest possible positive value. Shortly thereafter the values begin dropping and eventually go negative again.

Any sample you might examine will be composed of hundreds, probably thousands, of similar cycles. Now that we know what the data looks like in memory, let's examine file formats. You can look at any data file using the CLI command `TYPE <filename> OPT H`; the H option tells the type command that you want to see a hexadecimal dump of the file. This separates data into four four-byte groups of hexadecimal bytes per line, the equivalent ASCII characters, and a preceding offset address. We'll be looking at portions of actual sample files dumped using this method.

### Raw Data Dumps

The simplest way to store a sample is with a raw dump. There is no header, no overhead at all ... simply a "picture" of the data. Perfect Sound from SunRize Industries is capable of performing raw data dumps, and here is a piece of such a file:

```
0000: CCC8B7EF 33212900 A0ABAFCF 2F656770 ....31)....legp
0010: 40E4DDC8 BFEF3F3E 4B4400B4 C3CAD7FF @.....?>KD....
0020: 5F584A3F 28E09BAF BFCEDF37 6A615846 _XJ?{.....7aXF
0030: 38E0A0A3 B7CFD9EB 37645440 282310C0 8.....7dT@{#..
0040: 91A3BBD8 F852F6B 68514020 1004D090 ...../hO@....
0050: 93AFCFF7 0F18193F 6F6A5030 10F0DCD4 .....7qP0....
0060: B0949DBF ED0F2B3C 3C3F5F7B 64402000 .....<<<?_ID@.
0070: F4F4E0A0 8B9DB7D7 F7000F3F 7768543A .....?whT:
0080: 2A2800A8 97A7BDFE E7FF3F6C 54403534 '.....?IT@S4
0090: 00B0A7B3 C7C9D71F 4D3E3023 10C0A3BF .....M-0#....
00A0: DFEDFF4F 74605040 00A0979B A3BF1F3E ...OrP@.....>
00B0: 3D4630E0 B7C8BEF 3F645F50 00B0B2A5 =F0.....?D_P...
```

If you examine the center section, you can see the waveform moving through continuing cycles as described above. A quick glance at the right-hand section shows that nothing here is human-readable.

Now why would you want to store a sample this way, when IFF is the standard? After all, many programs won't even be able to use such a file. The best application for a raw data file like this is in dedicated programs of your own making. Suppose you write a game program that uses an explosion sound. You can write an IFF reader—in fact, there are complete IFF read/write routines available in the public domain—but if you only need to read in the same set of samples every time the program starts, this might be a bit of overkill. With a dump file, your program can simply read the sample directly into memory.

### IFF One-Shot Sample File Format

Now let's talk IFF. Most information in an IFF file (in fact, in most Amiga files) is stored in a "chunk"—a block of informa-

(continued)



tion with (usually) a header and trailer. The IFF format allows one chunk to exist within another, and requires multiple chunks per file.

Sampled sounds are currently recorded in an IFF format called 8SVX. Following is the first part of a simple one-shot 8SVX file:

```
0000: 464F524D 00003C88 38535658 56484452 FORM...8SVXVHDR
0010: 00000014 00003C08 00000000 00000000 .....
0020: 369C0100 00010000 414E4E4F 00000044 6.....ANNO...D
0030: 5265636F 72646564 20776974 68205045 Recorded with PE
0040: 52464543 5420534F 554E4420 66726F6D RFECT SOUND from
0050: 2053756E 52697A65 20496E64 75737472 SunRize Industr
0060: 6965732E 20202834 30392920 3834362D ies. (409) 846-
0070: 31333131 4348414E 00000004 00000002 1311CHAN.....
0080: 424F4459 00003C08 00010305 1F5F4428 BODY...<.....D[
0090: 10F41747 586F7067 69616866 6B7F7F7F ...Gopgjakfk...
00A0: 7F716F7E 7C605844 4F685036 3E373F6F .qo./XDOhP6-7?o
00B0: 7C605656 4C455750 302008F8 EAD7FB00 jVVLEWPO .....
```

Notice the first four bytes spell the word FORM. This indicates that we are looking at a simple IFF file, called a FORM chunk. (There are also LIST and CAT chunks, but we'll ignore them in this column.) Remember that each byte is represented by two digits in hexadecimal; so while FORM is only four characters long on the ASCII side of the printout, it requires eight characters (464F524D) in the hex portion.

Immediately following FORM, four bytes are used to indicate the size of the chunk; in this case, 3C88 hex or 15,496 bytes. This is the total count of bytes in the chunk excluding the chunk name and byte count; thus the file will be eight bytes longer than the value indicated. We'll see this pattern for each chunk.

In a FORM chunk, the next four bytes after the size indicate the type of IFF file. In this case, those bytes clearly spell out 8SVX.

Following the 8SVX are four more bytes marking the start of the voice header chunk, VHDR. This block contains special information about how the 8SVX data is to be interpreted. We'll take a look at the VHDR in just a moment.

Notice the letters ANNO after the voice header chunk. This is an optional annotation chunk, virtually unlimited in

size, which can contain any textual information one might want to include. In this case, it contains a short blurb for Perfect Sound, with which this particular sample was recorded. This ANNO chunk is automatically inserted by the program every time an IFF file is saved. An ANNO chunk is similar in structure to the FORM chunk mentioned earlier: the chunk name is followed immediately by four bytes indicating the size of the data (44 hex bytes), then the data itself.

After the ANNO chunk comes another named CHAN. This is in fact a non-IFF chunk, and it illustrates the beauty of the format. The Perfect Sound digitizer is capable of recording and playing back stereo samples. Although the IFF standard doesn't support stereo samples, its extensibility allows them to be accommodated. CHAN is a SunRize Industries chunk which specifies how the sample is to be assigned. Like the previous chunks, the CHAN name is followed by a four byte length, then the data. In this case, the data is a single four byte number. If the sample is to be assigned to the left channel, the number is two. It is four if the sample is assigned to the right channel, and six if assigned to both channels in stereo. Any program not expecting a CHAN chunk will ignore this data and load the sample normally. Similarly, Perfect Sound will load an IFF file without a CHAN chunk using a default assignment.

After the CHAN chunk comes yet another chunk named BODY, and as you might expect it contains the actual sample data. As before, a four byte length follows the chunk name; this particular sample is 3C08, or 15,368, bytes long. The actual data begins right after the byte count.

So, an IFF file (at least the ones we're concerned with) starts immediately with FORM and very shortly later identifies itself according to type. This can be handy when you have an unknown file, such as from a PD disk a friend has given you. If you can't tell if "BAG" is a bagatelle score file or a bagpipe sound sample, you can use TYPE OPT H to examine the first few lines of the file and find out almost instantly.

Notice that in this particular instance, a 15,368 byte audio sample file is preceded by 136 bytes of header information. In other words, the header comprises less than one percent of the total file size. Not only is this a pretty insignificant amount of overhead (which of course varies with the size of the sample and header), it is also important to reading an IFF file with a program expecting a raw dump file, such as the game program I used as an earlier example.

What would happen if you treated an IFF file as a raw data dump file? For the file in question (which has a sample rate of 13,980 samples per second), the 136 bytes of header information would last about 10 milliseconds if treated as sample data. This means that an IFF file can indeed be treated as a raw dump file, with the only negative effect being a very short click or pop at the start of the sample. In fact, the public-domain sample player program SSP uses this approach: it assumes everything in a file is data. Raw dump files play perfectly, and IFF files play with a brief click. (With SSP you can even play non-sound files like LIST or ED!)

### The Voice Header

Now back to the VHDR block I skipped earlier. As always, the first four bytes indicate the length of the chunk, and will always (at least until someone changes the specification) be 14 hex (20 decimal) for a VHDR chunk.

The next four bytes indicate the number of bytes in the "one-shot" portion of the sample—the non-looped portion you hear only once. This sample has 3C08 bytes in its one-shot section, which you will note is the same as the BODY length. This means the entire sample is a one-shot sound.

After the one-shot byte length comes another four bytes indicating the length of the repeating, or looped, section. You will notice that for our example sample this value is zero, confirming that this is indeed a one-shot sound.



Next are four bytes indicating the number of samples in the highest octave, a value which applies to IFF instruments. Since this is a one-shot sound, this value is also zero.

Then come two bytes specifying the sample rate, 369C hex or 13,980 samples per second. These are followed by a single byte indicating the number of octaves in the sample. Since this is a one-shot sound, there is only one octave.

The next byte in the VHDR indicates the form of data compression used. Since our sample is not compressed, this byte contains zero. We'll talk about data compression in more detail in a moment.

The final four bytes are a volume offset. The first two bytes are the whole portion of the number, the second two the fractional part. Here we have a volume level of 1.0, or normal full scale.

#### IFF Instrument Sample File Format

An IFF instrument file is closely related to the one-shot file we've just seen. Here, for example, is the start of such a file:

```
0000: 464F524D 00005DAA 38535658 56484452 FORM..8SVXVHDR
0010: 00000014 00000470 000008E0 00000020 .....P.....
0020: 29350300 00010000 4E414D45 0000000A J5.....NAME....
0030: 46757A7A 4775697A 61722863 29200000 FuzzGuitar(C) ..
0040: 00372053 6F756E64 2073616D 706C6520 .7 Sound sample
0050: 64617461 20636F70 79726967 68742028 data copyright (
0060: 63292031 39383620 456C6563 74726F6E C) 1986 Electron
0070: 69632041 7274732E 2000424F 44590000 ic Arts..BODY..
0080: 5D300000 FFFEF0FE FDFDF0FD FDFDF0FE J0.....
0090: FFF9FFFD FDFCF0C0 FDFDF0FC 00FEFCFC .....
00A0: F9FFFDFF FFFCFFFC FBFEF9FB FDFDFBFA .....
00B0: 00FEFFFC FAFEF9FF FEF7D000 FFFCF0FE .....
```

This is an instrument file supplied by Electronic Arts for use with Deluxe Music Construction Set. As you can see, it is closely related to the previous file. Note that there is no ANNO chunk. Instead, EA chose to use the optional NAME and "(C)" chunks to provide the sample name and copyright notice.

By comparing the VHDR chunk of this file with the previous example, we can see the difference between one-shot and instrument samples. First, notice that

instead of the entire sample being listed as one-shot, about one third of the length (470 hex bytes) is one-shot with the remainder (8E0 hex bytes) being looped. When this instrument is played, the player program plays through the sample and, upon reaching the end, continuously loops the second section.

Now look at the length of the BODY chunk, which is 5D30 hex bytes, a far cry from the 470+8E0 bytes just discussed. The reason is that this is an instrument file made up of multiple octaves—three, to be exact. The numbers given for one-shot and repeat sections always refer to the highest octave of a sample, which is the shortest and the first one in the BODY chunk. Immediately following this octave is the data for the next octave, which is exactly twice as long. In this manner, each octave is followed in turn by the next lower octave, with each one twice as large as the octave before it.

With three octaves, the total length of the BODY chunk should be  $(470+8E0)+2*(470+8E0)+4*(470+8E0)$  bytes. If you work this out on your HP16C calculator you'll come up with 5D30 bytes.

There is another difference between one-shot and instrument files. In a one-shot file the sample rate value is used to determine how to pace the playback. For an instrument file this number isn't normally used, since the playback rate is determined by the note to be played. Instead, the value for number of samples (data points) per cycle is used to calculate the proper playback speeds; for this instrument, the average number of samples per cycle is 20 hex, or 32. Now, if we divide the sampling rate (369C hex, or 13,980 samples per second) by the number of samples per cycle, we get 437 Hz. Since A above middle C is 440 Hz, it all works out pretty well.

(continued)

## TxEd PLUS<sup>tm</sup>

The Text Editor for the Amiga,<sup>tm</sup>  
Plus a whole lot more.

**BlitzDisk** Disk cache, speeds up floppy and hard disk reads up to 2000%.

**FastFonts** Speeds up text display.

**FunKeys** Hotkey window manipulator.

**ARP** Latest versions of the AmigaDOS<sup>tm</sup> Replacement Programs.

**AREXX** Demo version of the AREXX, the macro processor used by TxEd Plus, that is changing the way people think about computing.



Complete package:  
\$79.95

MC + Visa  
Mass Res. add 5%

### Microsmiths, Inc

PO Box 561, Cambridge MA 02140  
(617) 354-1224 BIX: cheath CIS: 74216,2117  
Amiga and AmigaDOS are trademarks of Commodore-Amiga, Inc



### Compressed Samples

We already know that a zero data compression byte indicates no compression. Although a large number of different compression schemes could be developed, only one method was defined at the time the IFF spec was written. That method, represented by a compression byte of one, is called Fibonacci Delta Compression. Here's how it works.

When dealing with a program such as ARC (which compresses program and data files), it is vitally important not to make any changes to the file being compressed. Change one bit in an audio sample, however, and the ear will not hear the difference. Change all the samples by a constant ratio, and nothing changes except the volume level. Clearly, we can make limited changes to an audio sample without damaging it, which provides much more flexibility when designing a compression scheme.

Fibonacci Delta Compression uses this fact to achieve maximum compression, and can best be understood by dissecting it in two steps. The first is to realize that what is stored is not the value of each sample, but the difference between samples, called the delta value or first derivative. In other words, if we had values of -3, -1, and 4, the delta values would be 2 (the difference between -3 and -1) and 5 (the difference between -1 and 4). We would prefix them with a starting value of -3, so we would know where to begin.

Using the first derivative of an audio waveform generally provides a smaller range of values, which are clustered about zero. Instead of actual values (which will range from -128 to 127), we are representing the rate of change, which will generally be much smaller.

The second step in Fibonacci Delta Compression is to approximate these rate-of-change values by sending not the values, but their locations in a table. In this particular implementation, the table contains sixteen entries, which can be represented by four bits each. Since the

data itself is eight bits, this means that there is a constant compression ratio of 50% (plus a two-byte overhead for the initial value and padding).

To cover the complete range of possible rate changes, we need to carefully select the contents of the lookup table. Here is where the "Fibonacci" comes in: our table is filled using the Fibonacci series (where each value is the sum of the previous two values: 1, 1, 2, 3, 5, 8, 13, and so on). In our lookup table the first "one" of the series is replaced with zero, and half the table represents negative slopes while the other half represents the positive ones. Notice that around zero the values will be very small and close together; this allows us to represent the smaller (and hopefully more frequent) slopes with accuracy. As the rate of change increases, our numbers spread out and we can only approximate the actual values, resulting in distortion.

How badly does Fibonacci Delta Compression affect samples? It depends on the samples. When applied to a fuzz guitar sound, the compression can be unnoticeable, because the sample is already highly distorted. When applied to a predominantly noisy sample (the whoosh of a grenade launcher or the sizzle of a high hat), this method seems to result in a loss of highs but causes little other damage. But when we try to compress the human voice or a flute, the result can be disastrous. Generally, the best approach is to try it and listen. If the sample is not seriously damaged, the compression can be used. Other than reducing disk space and improving loading speed, however, this form of compression has few advantages. Let's hope improved compression methods will be adopted for IFF in the future.

### Other Chunks

As you explore various samples with TYPE OPT H, you may find samples with nothing more than the minimal FORM, VHDR, and BODY chunks. On the other hand, you may see various combinations of optional chunks. We've already looked at ANNO, NAME, and (C) chunks.

Another you might see on rare occasion is the AUTH chunk, which gives credit to the "author" or creator of the sample.

The ATAK and RLSE chunks define the shape of the amplitude envelope. ATAK defines the sound from its start to the sustain level, while RLSE defines it from the end of the note to the final silence which follows. Both ATAK and RLSE chunks contain a list of straight-line segments defined by the duration of the segment and its ending volume. This is a feature of 8SVX which, surprisingly, has not been utilized often. There is quite a bit of power available here.

### For Further Study

This has been a brief—very brief—look at 8SVX IFF data files. With the information presented here, you can peruse a sample file, determine the format it is in, and perhaps even do a bit of "editing" with a sector editor. If you intend to write programs to read or write IFF sample files, however, you'll need much more information. One of the best sources is the *ROM Kernel Reference Manual*. The Exec volume contains a dissertation on the entire IFF standard (not just sample and score files) in Appendix B.

If you prefer electronic documentation (handy for searching with an editor) and happen to own a Perfect Sound digitizer, the complete IFF documentation is included on the program disk.

The next time you fire up your music program, take a few moments to browse through some samples first with TYPE OPT H. You never know what you might find.



•AC•



# Take Five!

## SUMMER ADVENTURES!

by Steve Hull

GENie: LightRaider

People Link: St.Ephen

### ROCKET RANGER

The year is 1940. The world trembles in the shadow of Adolf Hitler's 1000-Year Reich. Deep within the recesses of a top secret laboratory at Fort Dix, a young scientist puzzles over the physics of a high-altitude bombsight. His work is interrupted when a low, mysterious hum erupts in a blinding flash of energy, leaving behind five items from a future time—including a top-secret briefing manual. "A hundred years ago the Nazis won World War II," the manual begins. "It was a mistake. You can correct it..."

Talk about an attention-grabber. So begins *Rocket Ranger*, Cinemaware's flashiest, campiest, and indisputably best title to date.

As the story begins, a Nazi zeppelin lands in Washington, DC, where its crew kidnaps a leading American scientist and his beautiful daughter. Your first mission—if you're fast—is to intercept the zep as it drones back across the Atlantic. You strap on your rocket pack, dial up the correct amount of fuel on the Secret Decoder Wheel (a clever method of copy protection), and sprint to the parade grounds for your first takeoff. Getting off the ground is not as easy as comic books make it look, but after several attempts you'll get the hang of it!



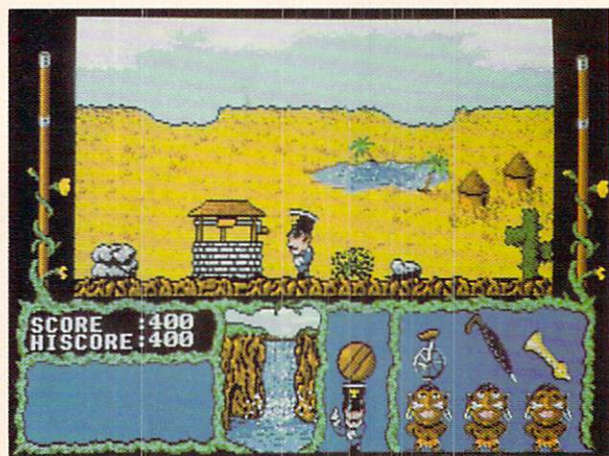
*Rocket Ranger*

The zeppelin rescue is just one of many subplots within the overall story, which concerns the Nazi discovery of a rare ore called lunarium. Lunarium is pretty handy stuff: you can make rocket fuel out of it, or bombs. A lunarium bomb has the effect of lowering the IQ of any male within the blast area by 30 points—after a bomb or two most men start acting like game show hosts (OK, OK, so I made that part up)—making the territory easy prey to invading Nazi hordes. Oddly, lunarium has no such effect on women—which suits the Germans fine, as they have special plans for them... BYOOO-hoohooohahahahahaaaaa!

There's only one small problem with lunarium: it can only be found on the moon. The captured women are subjected to treatment that turns them into zombified slaves, then shipped to a lunar base to work in the lunarium mines. This stuff is right out of pulp magazines—all we need now is an ad on the box for mail order bodybuilding courses.

(continued)





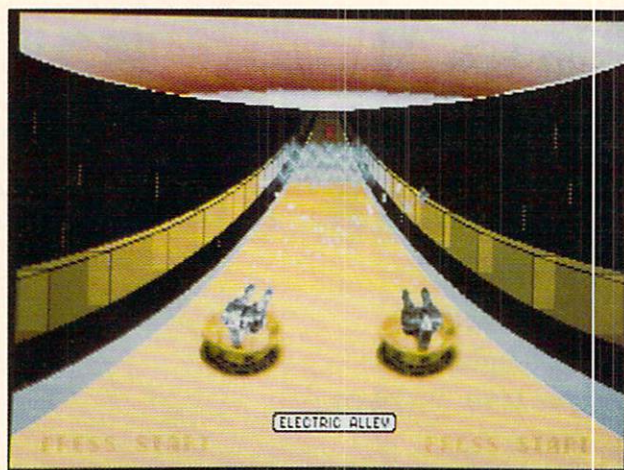
*Cosmic Relief*

In the course of thwarting the Nazi's conquest of the world, you will encounter the finest arcade sequences in any Cinemaware title yet. Among these: a challenging boxing match with a belligerent Nazi guard, shootouts with particle-beam weapons at a hidden jungle base, and the most original dogfight sequence you've ever seen—you, the Rocket Ranger, fly above rolling clouds taking on squadrons of German fighters! If you can steal the five parts it takes to build your own rocket and find fuel, more awaits you on the moon!

As easy as it is to get swept away by Rocket Ranger's graphics, the sound deserves special mention. The digitized sound effects throughout are very good, and the valiant theme music is outstanding. Perhaps the most striking feature is the characters' voices—actual digitized human voices, synchronized to the

characters' lips! The effect is so far superior to the Amiga's speech synthesis chips it'll make you want to open the case and pry the worthless things out!

Cinemaware programmers wrote their own disk operating system for Rocket Ranger, and it is reportedly three times faster than AmigaDOS. While there's no way to benchmark that from the game, it does load detailed graphics, animation, and sounds very fast. Since the game disks are not copy protected in any way, they may be copied to a hard disk or expansion memory. Cinemaware President Bob Jacob says Rocket Ranger actually runs faster from its own floppy disks than a hard disk, though running the program from expansion memory offers a slight increase in speed. It is possible to play the game with only one floppy drive, but Cinemaware strongly discourages this, and with good reason.



*Roadwars*

This column makes 35 games I have reviewed this year, and I have evaluated many more than that. Looking back on this review I am struck by the number of exclamation points I have used—Rocket Ranger is just that kind of game. It is the most impressive title I have seen on the Amiga yet—this year's must-have!

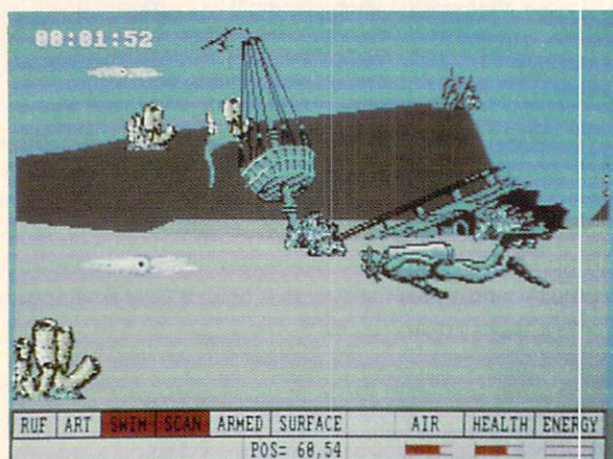
#### **COSMIC RELIEF**

The year is 1948. Rocket Ranger has booted the Nazis off the moon, and America is endeavoring to forget World War II. The accent is on the positive as the country focuses on the bright future ahead. No one is particularly interested in a certain eccentric Professor K.K. Renegade, or his predictions of an asteroid on a collision course with the Earth. "Have a Coke and a smile, Renegade," the crowd sneers. Such are

*(continued on page 36)*



*Sub Battle Simulator*



*Return to Atlantis*



# Hot on the Shelves

by Michael T. Cabral and Michael Creeden

## Strategic Warfare

The imperialistic Krellan Empire is invading Alliance airspace, swallowing up planets like dots in a Pac Man maze. Through Operation Big Brother, the Krellans use the conquered planets to strengthen their empire. They transport strategy and production teams to each planet's largest city, then install a Krellan general as emperor. Using the invaded planet's resources—both natural and human—the Krellans begin their conquest. When it is completed, the Krellans add yet another planet to their ravenous empire.

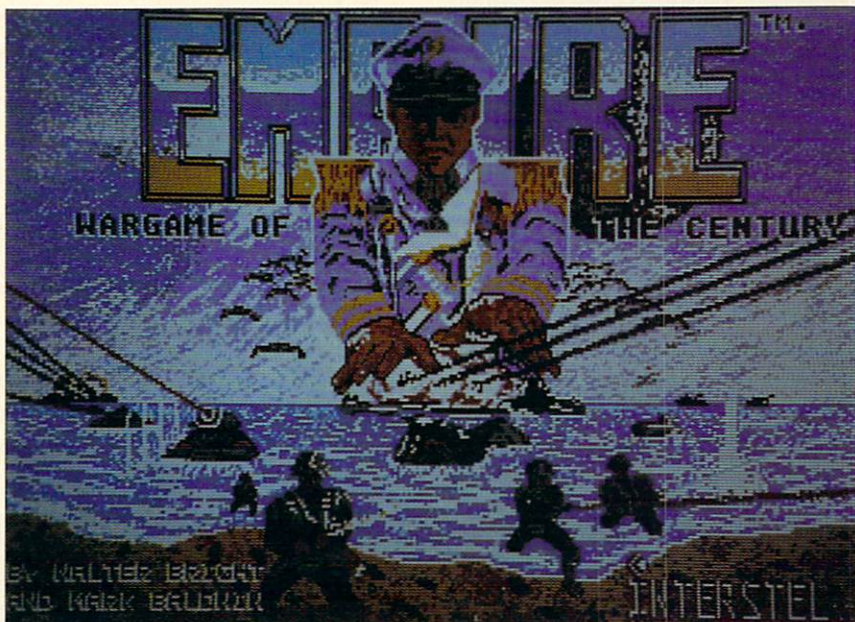
Time is running out. You, Captain William P. Brown, and your crew have a mission: stop the Krellan monster from achieving its goal of total domination of Alliance air space. You must patrol Alliance space and intercept the Krellan vessels as you attempt to stop Operation Big Brother. You plan to beam down your advisors to the Krellan-invaded planets and attempt to unite the servile citizens. Your advisors will form a counter-empire to defeat the growing Krellan empire and convert the planet to democracy. Your goal is also total domination of Alliance air space.

Interstel's **Empire** is a strategic simulation featuring global conflict, conquest, and empire building for one to three players. You begin with one city on a large, unexplored planet. Your goal is to capture cities that produce the war materials you need to defeat the Krellan Empire.

That's where the strategy comes in. As you conquer more cities, you must decide what each city should produce to help aid the war effort. As your planetary war chest grows, you command and maneuver armies, fighters, and ships to crush the Krellan Empire and quench its voracious appetite for land.

Empire features land, sea, and air combat. Your fleet is comprised of your choice of aircraft carriers, battleships, cruisers, destroyers, submarines, and troop transports—all you need to conquer the Krellans and make the galaxy safe for democracy.

But don't breathe too easily. Your opponents have fleets of their own, and they're out there somewhere. The key word is "somewhere." If you are looking for a good ole' shoot'em up time, you had better look elsewhere. Empire is a strategic simulation, not an arcade game. Arcade fans may find Empire more of a challenge to their patience



Empire title screen



than to their gaming skills—up to 200 turns of play can pass before you even spot your enemy! Once you find your enemies, you must defeat them through shrewd planning and brilliant tactics, not quick reflexes and hand-eye coordination. For strategy buffs, Empire offers a feast of brain-wrenching fun.

### **Empire**

\$49.95

Electronic Arts

1820 Gateway Dr.

San Mateo, CA 94404

(800) 245-8525; in CA (800) 562-112

### **Joyous Gaming**

With all the arcade-style titles now on the market, Mindscape, Inc. has decided to support the aching hands of all their faithful fans. Now on the shelves is the **PowerPlayers Joystick**, a treat for failing fingers and calloused palms everywhere.

The most visible difference between this stick and the scads of others on the market is that the PowerPlayers is meant to be hand-held. Many enthusiastic gamers have become disenchanted with flat-bottomed joysticks that require a tabletop. This new stick lets you really get into it. You can hop around, speak your best body language, and turn with

the turns. The pistol-like handle design with deep finger grips gives you a real feel for the action in either hand. An extra long cord gives you even more freedom to romp around as you get swept into the action.

Playability, though, is the deciding factor for a joystick. A stick can feel great in your hands, but if the mechanics are sluggish and glitchy, gameplay becomes misery. A gamer's worst frustration is having his skills doused by an unresponsive stick. The PowerPlayers uses a steel shaft and ball bearing pivot for reliable response. The stick has a nice, easy click to it. Wearability is also taken care of by the steel shaft. If you break off this handle, you're probably taking the game you're playing a little too seriously!



*The PowerPlayers Joystick*

The trigger, the other half of the playability team, is a hearty, red button at the top of the handle. With the PowerPlayers, you don't have to worry about clumsily missing the fire button at a deadly moment. When you wrap your hand around the stick, your index finger rests comfortably on the trigger. Ultra-sensitive microswitches make firing rapid. If you miss a shot that gets you burned, you've got nobody to blame but yourself!

Mindscape has also set up the PowerPlayers Club, a fraternity for incurable gameaholics. For the price of an SASI, you can become a member and receive newsletters, hot Mindscape product announcements, and cash discounts. The premiere issue offers the PowerPlayers joystick for five bucks off the retail price. Not bad for the price of a stamp.

### **PowerPlayers Joystick**

\$29.95

Mindscape, Inc.

3444 Dundee Rd.

Northbrook, IL 60062

(312) 480-7667

SoftWood File IIsg			
MAGDEX			
	TITLE	KEY	AUTHOR
364	Analyze! V2.0	Spreadsheet	Schaffer Kim
365	Impact - Business Graphics	Business	Raudonis Chuck
366	Microfiche Filer	Database	Laser Harv
367	PageSetter	Desktop Publ	Winch Rick
368	Giznoz Productivity Set V2.0	Toolkit	Eller Bob
369	KickWork	Kickstart	Laser Harv
370	Diga! - TeleCommunications Package	Terminal Pgm	Hull Steve
371	Bill Volk - Aegis Vice President	Personality	Hull Steve
372	Time Saver	Clock	Foust John
373	Mouse Time	Clock	Foust John
374	Insider - Memory Expansion Card	Memory Exp	O'Keane James
375	Microbotics Starboard 2	Memory Exp	Falwiczewski Ste
376	AnigaNotes	Music	Rae Richard

*A typical Magdex listing.*



### Sort Your Resources

You saw a great little utility somewhere in your favorite Amiga mag sometime in the past two years. Somewhere and sometime are the key words, though. You have no idea when or where the article appeared. Your only solution is the frustrating, time-gobbling task of sifting through all the magazines in your bookcase. If you stumble upon the article you're looking for, you can sit back and be temporarily pleased with your inefficient system and stroke of luck. But with a powerful data management computer like the Amiga sitting stagnant on your desk, isn't it foolish to thumb through pages and pages to find needed information?

Amiga enthusiast Stan S. Spence has a solution to the page-flipping foolishness with **Magdex**, a databased catalog of seven major Amiga publications. Amazing Computing, AmigaWorld, Amiga Transactor, Commodore, Commodore Power/Play, The Amigan, and Transactor are all covered. And, since you can get the runaround with the public domain disks out there, **Pubdex** sorts out over 2200 PD programs. Unmasked are the Fred Fish and Amicus collections and more. As more mags and PD programs hit the market, you're free to add them. The databases are based on a demo version of Softwood File IISG.

When you boot Magdex or Pubdex, the delightfully carefree Looney Toons theme chimes and the screen colors cycle. What seems like a cute way to pass the boot time is actually an anti-virus program called BootTune. If there's no music or color cycling, you're probably in trouble.

Across a spreadsheet, Magdex lists generics like magazine and article titles, along with the year, volume, issue, month, and page. A Type column lets you know whether the article is a review, tutorial, program, table, or whatever. If you want to know specifically what an article is about, check the Key column. Here you find distinctions ranging from sports games to AmigaDOS, from video to Forth. If you still need more information, a Comments column lists any other pertinent notes.



*The Viking 1 monochrome monitor*

Similarly, Pubdex lists the program name, version, author, date, and the disk number and collection. A Category column splits programs into ray-tracing, graphics, games, etc. Especially useful is the Comments column which tells you what a particular file or program does.

Best of all, you can put all that wasted disking-combing and page-scouring time behind you for only ten bucks.

#### **Magdex, Pubdex**

*\$5 each*

*Stan S. Spence  
5147 S. 37th St.  
Lincoln, NE 68516  
(402) 423-3856*

#### **Zap Spelling Zingers**

Shouldn't a spelling checker do more than just keep you from looking dumb? Anybody can scan a finished document and punch the keys to replace misspellings. Ideally, a spelling checker should make you a better speller by actually teaching as you go. Meridian Software Inc. has come up with a possible route to the ideal with their latest release, **Zing! Spell**.

Interactive checking through multitasking is the key to Zing! You can check your spelling and patch your blunders as you type. This way the botched word is fresh in your mind in context, along with the correct spelling. The next time you start to tap that word in, you will probably remember the slip you made last time. If you don't want to be pestered after every word, Zing! also lets you check after each line or paragraph. If you want to toss out the interactive advantage to save time or whatever, a batch option lets you fly through the entire document.

The main window lets you in on the last misspelled word and the total number of words hacked so far. A window drag lets you see the entire list of boo boos and allows you to cut and paste computer-specific, unrecognized terms to a custom user dictionary. The main dictionary is packed with 95,000 terms, and memory permitting, you can add anything from 95,001 on up to your user dictionary.

A gadget click (or one of many key commands) brings up the spelling help window with the misspelled word and a selection of suggested spellings. Another click replaces the erred word with any of



the suggestions. Wholesale substitutions can be accomplished with the exchange window, a search-and-replace clone. Type a symbol or single letter of an oft-used but bulky word, and you can insert the entire word throughout the document in one fell swoop.

Zing! Spell has other niceties in the package, including full AREXX support. If your eardrums and nerves have had enough of the beep that indicates an error, you can toggle it off. Send in your registration card and you get a free thesaurus. With flawless spelling in-hand and a better vocab on the way, you can spell literary success Z-I-N-G!

### **Zing! Spell**

\$79.95

Meridian Software, Inc.  
9361 West Brittany Avenue  
Littleton, CO 80123  
(303) 979-4140

### **Monochrome Mastery**

Commodore recently announced the introduction of **Viking 1**, a 19-inch high resolution monochrome monitor for the Amiga. A late August time frame was tentatively set for release of the monitor.

The monitor is the product of a joint development between Commodore and Moniterm, a leading manufacturer of high resolution monitors. The Viking 1 is designed to expand the Amiga 2000's reach into the graphic workstation market—a market that supports applications like desktop publishing, CAD/CAM and graphic illustration.

The Viking 1 features a 1008 x 800 x 2 bit resolution, 72 MHz pixel frequency and 56 KHz horizontal frequency. The monitor has a 19.6 inch screen and weighs 36 pounds. Its overall dimensions are 17.8 x 14.5 x 15 inches.

Commodore president Max Toy said "the development of such a sharp, high-resolution monitor as the Viking 1 expands the A2000 to even higher-level graphics applications and is a natural extension of our overall focus on the graphic workstation environment." Toy

called the Viking 1 "another example of the expanding commitment third party developers are making to the Amiga."

The Viking 1 will be distributed by Moniterm with a retail of \$1995. The monitor will be on display at Siggraph in the Commodore booth, #1426, and the Moniterm booth, #2150 at the Georgia World Congress Center in Atlanta.

### **Viking 1**

\$1995

Moniterm Corporation.  
5740 Green Circle Dr.  
Minnetonka, MN  
(612) 935-4151

### **Lights! Camera! Desktop Presentation!**

Many useful applications are no more than new combinations of what's already been accomplished. Desktop presentation fits in here as an application that has taken a little bit from everybody to become a viable independent entity. Genlocking, titling, animation, advanced graphics, IFF sampled sound, and other Amiga wonders have all chipped in to carve a market for desktop presentation.

The latest name to enter the cozily small fraternity of desktop presentation developers is Aegis Development, Inc. with **Lights! Camera! Action!** by Sparta, Inc. Authored by Gary Bonham, crafter of the ANIM format, LCA lets you join IFF pictures with ANIM-based animations and Sonix format samples to slap together knockout presentations.

Any animations created with the ANIM format, including stuff from Aegis's VideoScape 3D and VideoTitler (Why not support your own, right?), and The Director by The Right Answers Group are fine by LCA. All Amiga screen resolutions can be handled, and interlace and overscan are supported in any resolution. If you want to put your finished presentations on a video tape, overscan is crucial. If video tape isn't the route you want to go, ShowLCA, an independent program included here, lets you save your presentations to disk and distribute them without the entire LCA program.

LCA provides over forty options for the bits of flash you need for transitions. As you move from one image or animation to the next, LCA lets you incorporate wipes in nine directions, dissolves flips, fades, and more. Multiple block effects like burst, zig zag, checker, spiral and wrap can help your presentation look more and more like something straight out of NBC studios. A Multiple View Port feature lets you scroll, cover, reveal or wipe as the next animation in line takes over the screen.

LCA supports both entire musical scores and IFF samples in Sonix SMUS format. Whether you want a running musical background or just certain sound effects for certain frames, LCA supports your whim. You have control over the audio, too. You specify the precise frames you want the sound to start and stop on, and pitch, timing, and volume control are yours at the click of the mouse.

LCA also gives you complete control of your presentation at playback. You set the timing for all frames, music, and animations and have manual, single, and looping modes at your disposal. As other advanced features go, LCA also supports HAM and the extra halfbrite chip. Full software support is included for the SuperGen genlock. And if you're from across the Atlantic or the gorgeous land "down under," LCA complies to the Australian and European PAL standards.

You need at least 1MB to run LCA, and two or more megs are recommended if you hope to stretch the program to its limit. To get you started, a disk of sample presentations, IFF shots, and music is included, along with Grab ANIM, a screen grabbing utility. Want your flashiest ideas to look just as good on-screen as they do on your mental screen? Try Lights! Camera! Action!

### **Lights! Camera! Action!**

\$79.95

Aegis Development, Inc.  
2115 Pico Blvd.  
Santa Monica, CA 90405  
(800) 458-5078; in CA (213) 392-9972

•AC•



## *Easedrop with EaseWare.....*

*Listen in on a HomeBuilders\_CAD users telephone conversation.....*

*Sam: Bob, this is Sam. I bought that HomeBuilders\_CAD program you're so gung ho about. What's with you? It doesn't have one chimney! Have you seen any houses going up around here without a chimney?*

*Bob: Sam..... Sam.....*

*Sam: It doesn't have fireplaces, no columns, and I can't draw shrubbery...*

*B: Sam, I know how you can draw those.*

*S: And what about furniture? You would think a program like HomeBuilders\_CAD would let me add furniture.*

*B: Sam, if you'd be quiet for a minute, I could tell you about the bow window I added and how I changed the joist sizes.*

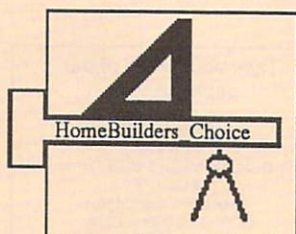
*S: I want that flexibility too, but it doesn't work! Are we talking about the same program?*

*B: Sam, shut up and listen! OK? EaseWare had to start somewhere. Users like me wanted their own custom parts library and the ability to change HomeBuilders\_CAD building parameters. EaseWare listened to us. I just bought HomeBuilders\_Choice. It lets me create any part I want and use it with HomeBuilders\_CAD. I designed some California closets for that client who just moved to Boston. The guy is satisfied now. Funny that closet interiors were such a big deal but they got me the contract.*

*S: I'm convinced. I'll buy HomeBuilders\_Choice, but I still can't use my plotter with HomeBuilders\_CAD.*

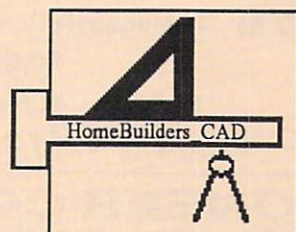
*B: In October you can...*

## *EaseWare Announces HomeBuilders\_Choice* *The build your own custom parts library* *program for HomeBuilders\_CAD.*



Suggested retail price \$81.00 USA

**EaseWare**  
25 Belair Rd.  
Wellesley MA.  
02181  
617-237-2148



Suggested retail price \$199.00 USA

These EaseWare products require an Amiga with 1 megabyte and 2 disk drives  
Amiga is a registered trademark of Commodore-Amiga



*(IntelliType, continued from page 8)*

Programmer Moses Ma spent a long time gathering data about speed-reading, brain chemistry functions, and neuro-physiology before creating IntelliType. A graduate of Cal Tech and a student of such diverse fields as meditation and martial arts, he learned how the mind can be stimulated by reactions to stress and increase its ability to accept new challenges and information. Ma's research into how people learn has enabled him to create a typing tutor that helps them learn.

IntelliType uses an advanced programming technique known as "artificial intelligence" to adapt to the user. The more you work with IntelliType, the more it knows about your quirks and bad habits. With this information, it can retrain your mind and hands to become better and faster. It really works.

#### ***It Teaches At Your Own Pace***

IntelliType sets up a 30- or 60-day typing class, depending on how much time you

want to devote each day. The program can be booted off its non-protected disk (kudos to Electronic Arts for finally removing copy protection from its Amiga productivity software!) or, memory permitting, you can multitask it with your other software.

IntelliType wants certain directories assigned to its disk when you start the program. Rebooting on the disk makes this automatic. I used a public domain program called "Defdisk" to perform six reassign commands to the target disk. I could then pop IntelliType into my drive and run it from its icon, and still flip back to my other running programs, my Workbench, or CLI. When I finished with IntelliType I just ran Defdisk again and reassigned all my system directories back to my regular SYS: disk.

Two floppy drives are helpful, since IntelliType wants to save reports on your typing progress to a "Student" disk. If you have only a single floppy drive, some disk swapping is necessary.

Although IntelliType can be used by someone like my friend Jim who has no typing skills, it can also adapt itself to an intermediate or advanced typist. It helps experienced typists recognize their rough spots and emphasizes lessons that help overcome years of bad typing habits.

IntelliType's 25-page manual is well written and a good introduction to the program's features. Since IntelliType is sublimely easy to learn to use, you probably won't need to look back at the manual initially after reading it.

IntelliType uses an easy-to-read font on a low-resolution screen, with a comfortable color scheme similar to words on a paper page. It's a pleasing environment that keeps you coming back.

Daily lessons begin with comments on your past performance and some simple warm-up drills, just as if you were in a "real" typing class. Windows at the bottom of the screen gauge your average speed and accuracy. If the windows distract you, you can remove them.

## **PIONEER COMPUTING Does it again!**

### **PRD-44**

#### **44 Megabyte Removable Cartridge Winchester Disk Drive**

The PRD-44 Cartridge Hard Disk is an innovative design that incorporates reliable Winchester Technology in a removable media. You will never outgrow this Winchester as the PRD-44 offers you unlimited storage. The PRD-44 has many benefits, including low cost, mass storage, back-up capability, transportable and rugged cartridge media.

At 44 megabytes per cartridge, individuals may maintain large amounts of data for individual or share system applications. The compact 5 1/4" cartridge permits data security as users may remove and secure sensitive data.

The drive has an average access time of 25 milliseconds and a 1:1 interleave capability. Within 20 seconds after cartridge insertion, the PRD-44 has completed spin-up, self-test diagnostics and is ready to accept commands. Additionally, automatic error detection/correction and extensive defect management are transparent to the user.

#### **PRD-44 44 Megabyte Removable Cartridge Disk Drive**

**\$999.99**

#### **PRD-44c 44 Megabyte Cartridge**

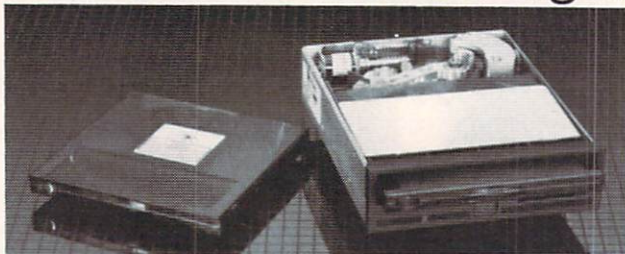
**\$129.99**

### **PIONEER COMPUTING**

2469 East 7000 South #200 - Salt Lake City, Ut 84121

(801) 942-1174

ORDER DESK 1-800-999-3013



#### **KEY FEATURES:**

- Removable Cartridge Media
- 44 Megabyte (formatted) Storage Capacity, per Cartridge
- Unlimited off line storage
- Fully Integrated SCSI Controller
- 25 Millisecond Average Access Time
- Standard Half-Height 5 1/4" Size
- 20,000 Hour MTBF
- Data Transfer Rates up to 1.25 Megabytes per Second
- 8 K Data Buffer
- Read/Write Operations with 1:1 Interleave
- ECC and Automatic Retry
- Transparent Defect Management with Track and Sector Sparing
- Self-Diagnostics at Power Up
- Rugged Hard Disk Cartridges

#### **Take advantage of our package deal:**

Disk Drive & Cartridge  
\$1049.99  
Disk Drive, Cartridge & SCSI Controller  
\$1279.99  
Disk Drive, Cartridge,  
SCSI Controller & Case  
\$1399.99



### ***Romance! Intrigue! Adventure! Typing Lessons?!***

After warming up, you're ready to move on to the day's lesson. Since IntelliType is billed as a typing tutor for adults, you follow the exploits and romance of Ted and Laura through a novella. Along the way, they encounter international spies and intrigue while searching for a secret formula. The prose is well written and captivating—a far cry from the boring typing lessons of my high school days. Each episode of the story ends with a “cliffhanger,” encouraging you to come back for the next lesson.

The exercises are mapped out as triple-spaced lines of text. Your typed input appears a line below what you're reading. Audio and visual prompts (which may be toggled off) alert you when you make a typing error. When you complete a screen, the next one appears automatically. The environment is very comfortable and has a good, responsive feel to it.

While you're working on your lessons, IntelliType is busy “behind the scenes” compiling statistics on your progress. This process is completely transparent; you never know it's happening. The program knows all the common typing errors and many peculiar problems (such as dyslexia, where the eye sees the set of characters, but the brain translates them incorrectly). IntelliType knows triplets, trigraphs, alternates, hand balance, doubles, transpositions, wrong-hand shifting, unnecessary shifting, and on and on—nearly every kind of typing error.

IntelliType monitors your progress, and makes suggestions for repeating drills or passages of text which give you trouble. You can configure the program dozens of ways (via drop down menus) to work with you. Tell IntelliType to emphasize typing speed or accuracy or both. Ask it to exert more psychological “pressure” during speedtyping lessons and it forces you to look ahead rather than look at letters as you type them. This is a common technique in speedreading instruction, too.

### ***Hi-tech Analysis Of Your Progress***

When you finish your lesson for the day, a series of fancy 3-D bar charts and explanatory texts show the results of the in-depth analysis IntelliType performs while you enjoy Ted and Laura's latest escapades.

You see exactly what your strong and weak points are—all those possibilities I mentioned earlier. While you are typing, IntelliType even counts the microseconds between your keystrokes and knows which keyboard areas give you trouble! Again, you don't start from scratch each day; the strategy for each lesson is based on your previous day's performance. Your personal data is stored on that “Student” disk and fetched by IntelliType. This is a smart program; you really feel as if it knows how *you* type and knows exactly where you need more help. IntelliType is not only very smart, it's very solid too. It never crashed once while I was using it.

## **Erich Stein & Associates, Inc.**

Public Relations Consultants

***Because the quality of your  
reputation is just as important  
as the quality of your product.***

PO Box 695

Denver, Colorado 80201

TEL (303) 733-3707

I consider myself an advanced typist with good all-around keyboard competence, but IntelliType helped me to correct some sloppy habits that I hadn't even noticed. If your typing skills aren't quite up to snuff, or you really want to learn to type well, I can't think of a better or more enjoyable way.

•AC•

### **INTELLITYPE \$49.95**

*Not copy protected*

*Requires OS 1.2, 512K, one disk drive  
and one blank formatted disk. Second  
drive helpful but not required.*

*Electronic Arts*

*1820 Gateway Drive*

*San Mateo, CA 94404*

*(415) 571-7171*

### **About The Author:**

*Harv Laser, 38, is the Senior Chairman (Sysop) of PeopleLink's AmigaZone Club and an Amiga consultant. He has worked with the Amiga since 1985 and his writing has appeared in INFO, Amazing Computing, AmigaTimes, and AMnews.*



(*Take Five*, continued from page 28)

the problems faced by great minds. The professor tramps off into the sunset, muttering arcane formulas under his breath.

But wait! A giant space rock the size of Louisiana has been sighted hurtling toward Earth! The call goes out—find Professor Renegade!

Such is the premise of Datasoft's *Cosmic Relief: Professor Renegade to the Rescue*, marketed by Electronic Arts under their Affiliated Labels program.

In *Cosmic Relief*, you direct one of five intrepid explorers through some of the strangest terrain yet encountered in a video game. Your search for the Professor will lead you over the clouds, riding an airborne vacuum cleaner ("special attachments," the documentation explains), through caverns, and even across a tightwire on a unicycle. Stone snakes will menace you, as will flying lizards with little Chihuahua faces. Wild Kingdom was never like this.

Along the way you will encounter tools and other items: a coat hanger, a scroll, a flash camera. Some, like the Acme Expanding Bridge, are imperative to continuing your search; others, like the teacup and the 9-volt battery, will be needed to construct the asteroid deflector (but you knew that, right?).

The game's graphics are humorous and well-done, and while the onscreen explorer is generally responsive to your commands, sometimes he will balk at especially stupid moves. Try to walk off a cloud, for instance, and he will shake his head violently. The sound effects are pretty simple, and there is a nice stereo soundtrack that can be turned off when it starts driving you wacky.

Achieving success takes ingenuity, reflexes, and no small amount of luck. Part of the challenge is figuring out which item to use to get past an obstacle. You use the flute, for example, to charm the snake that will otherwise turn your onscreen player into Explorer McNuggets. If you get stuck, pressing "T" prompts your player to think. Sometimes

this produces only a shrug and much head shaking; other times he will nod excitedly, pointing finger to sky as the suggested item appears above his head.

If all else fails, you can shamelessly break the seal on the Cheat Notes and Map that accompanies the documentation. Rumor has it that these notes contain a chart of the entire multilevel play area, and a list of items and their purposes. I, of course, would never stoop to using such notes (not much).

In testing *Cosmic Relief*, I discovered the game's programming is a little finicky about keyboard input. It is unplayable on early Amiga 2000's with the "German" keyboard, and also balks at 1000's equipped with C Ltd's Timesaver. Besides the software glitches, there is one apparent design bug. The documentation mentions your time is limited to 30 "days" before the asteroid smashes into the Earth. However, finding out the amount of days elapsed is a puzzle in itself. It's in the game, but not displayed as prominently as one might expect.

"*Cosmic Relief* promotes critical thinking and strategic planning skills," the game's press release says. Right. Kids, remember that line the next time you're in the software store. "But Mommmmm, it promotes strategic planning skills!"

Sure, kid. Just don't let daddy try or you'll be stuck watching *Cosby* reruns on the VCR for the night. *Cosmic Relief* is addictive, challenging, and a lot of fun.

#### ROADWARS

It took 25 centuries A.D. to do it, but intergalactic wars have exiled mankind to the moons of the planet Armageddon. In a feat of engineering virtuosity, the moons have been linked by a network of space roadways, controlled by computers which maintain the links despite the fact that the moons orbit Armageddon at different speeds!

But something has gone wrong. The magnetic panels created to gently guide traffic have begun shooting violent blue sparks; treacherous spikes have appeared

on the roads, and killer satellites are vaporizing interlunar commuters. Something has subverted the computers, turning mechanisms created to protect and enhance life into something irrational and destructive! (Anyone who has ever managed a local area network will recognize the phenomenon immediately.)

In *Arcadia's Roadwars*, you take control of a battlesphere assigned to rid the spaceways of lethal hazards. The press information I received describes the battlesphere as an "invincible" interplanetary tank (obviously, whoever wrote that never played the game—my battlesphere was extremely vincible). Clear the road!

You begin a level paired with a second battlesphere, controlled by either the computer or another player. The battlespheres roll side-by-side at a constant speed along the track that circles the moon. Working in partnership, you and the other battlesphere must neutralize the rogue blue panels which send deadly sparks arcing across your path, plus a collection of other nasties, such as red energy balls that run counter to you, and the periodic killer satellite that is guaranteed to ruin your day if you don't blast it to tinfoil first.

The battlespheres are protected by shields which must be dropped to fire weapons. The shields are resistant to sparks, but impact with other enemies will destroy the shields, leaving the battlesphere defenseless.

*Roadwars* lacks some features that make home game playing more enjoyable. There is no way to pause a game, no small drawback when you're into five levels and the phone rings. If you get off to a bad start, the only way to abort a game in progress and restart is to use the kamikaze method—that is, kill off your Battlesphere enough times to get a GAME OVER.

There are other glitches as well. High scores are not saved to disk; when you play against the computer, the computer's player doesn't even keep score. There is no way for advanced



players to begin at more demanding skill levels. The game's "help" screen—15 lines' worth—flashes up for only six seconds during the game's attract mode. Hasn't anyone noticed the Amiga keyboard has a key labeled "HELP"?

As is typical of Arcadia's titles, Roadwars' graphics, sound, and music are very high quality. Gameplay is fast, original and addictive. Despite this, a little more attention to detail would be welcome. Arcadia needs to pull this baby back into spacedock for a tuneup.

### **SUB BATTLE SIMULATOR**

And now for a short pause, amid all the raucousness and swashbuckling, for something nice and cerebral... like blowing a few thousand tons of ships out of the water!

Submarine simulations, like flight simulations, have traditionally caught the imagination of computer gamers in a big way. The genre started simply, as mere seafaring shooting galleries whose sole test consisted of picking off ship silhouettes that steamed like target ducks across a round periscope's-eye view. Such 8-bit classics as Silent Service and Gato added new levels of detail, ultimately selling well enough to earn translation to 16-bit architecture.

Sub Battle Simulator is Epyx's latest entry in the proverbial submarine races. A translation from the Macintosh, it approaches the theme from a strategic, rather than an action, orientation. As one Epyx staffer observed, "this is not a game that you will sit down and blow away in a few minutes." If you're looking for arcade action, skip to the next review.

On the other hand, if you're looking for a historical simulation that will stretch your intellect, step right up. Sub Battle Simulator lets you choose between submarines of the American and German fleets that prowled the seaways between 1939 and 1945. You begin with simple target practice against an enemy convoy, graduating to missions based on actual WWII scenarios. For the ultimate test, selecting Wartime Command takes you

through the entire war! Successfully completing a Wartime Command earns you a place in the disk's Hall of Fame, but the operative word there is earn. Fortunately, Sub Battle Simulator will let you save a game, to continue later.

A wide range of options allows you to handle details yourself, or delegate to the computer. At lower levels, selecting Navigator allows the computer to chart your course. In any phase of the game, time may be compressed as much as four hours per second, or played in real time.

The main documentation, written for the Macintosh, is supplemented by two quick reference cards for the Amiga. Each card is packed with options. You may steam along on the surface under diesel power, or submerge and switch on the electric motor for silent running. You have both sonar and radar, plus your own eyes' view from the tower, periscope or binoculars. Torpedoes, mines, and deck guns are available to raise havoc with your prey.

Of course, most of the prey out there shoots back, so look sharp, Cap'n. As the documentation explains, the ships' larger and more stable guns allow them to hit you before you can hit them. You will also find yourself fending off attacks from hostile aircraft, as the displays in your status indicator rapidly flash from green to yellow, then to red!

The submarines' most effective strategy was stealth—waiting until dark to surface and attack. The documentation offers much in the way of strategy and training, making the game easy to learn—but as in the real world, difficult to master.

Give Epyx credit for producing a class act with Sub Battle Simulator. It provides a thought provoking and thorough look at the lives of those who served in the silent service.

### **RETURN TO ATLANTIS**

To call Return to Atlantis "long awaited" is to exercise extreme understatement. Electronic Arts actually announced the

game prior to the Amiga's launch, in the summer of 1985. I still have a copy of that ad: EA President Trip Hawkins, resplendent in his preppy pullover, facing a page of real-soon-now titles. Among them, Return to Atlantis, a game promising to put you in the shoes (or wet suit, as it were) of "Indiana Cousteau, oceanic hero."

Anyone who has played Return to Atlantis will attest that the slant of the game has changed quite a bit since the days of Indiana Cousteau. Its slow metamorphosis from Raiders of the Lost Joystick to Flipper Goes Greenpeace has provided some of the most intriguing gossip (and leaked betaware) the Amiga community has seen in a long time.

If Cinemaware has taken the metaphor of the big screen, Return to Atlantis takes the approach of the small one—the game is made up of fourteen intriguing "episodes," much like a TV series. Each episode has a specific goal for you to achieve. You will quickly learn that unlike a TV series, these episodes are not easily wrapped up in 30 minutes.

You begin Return to Atlantis by creating a character. This process consists of entering a name, age, and sex, though this is about 95% vanity. Two test characters I created—an 80-year-old woman named Gertrude and a 5-year-old boy named Skippy—played identically. This done, the computer assigns numeric values to such character attributes as psychology, weapons, and telepathy.

After a short mission briefing by a holographic version of the standard Amorphic Superior Being, you are ready to begin. First stop: the infamous Sea Thief Cafe.

At the Sea Thief, you may beg, buy, or threaten five informants for additional details to assist in your quest. This sequence features the best graphics in the game, with each character lushly rendered. You select a questioning strategy by listening to the informant, then clicking on PLEAD, BRIBE, THREATEN, YES, NO, or GOODBYE,

*(continued)*



**MUSIC VISIONS** ©

Create a light show on your Amiga. Music Visions adds the visual dimension to music as it plays, using the Perfect Sound or FutureSound digitizers. Music Visions samples and analyzes the music, producing a light show in real-time!

Create a show to your taste by using a drawing program that creates IFF files we call musicfiles. Musicfiles can be a simple or complex: the only limit is your imagination! Options allow you to change the show as the music plays.

- Bi-directional, variable speed, color cycling
- 3 display modes • Uses any lo-res IFF picture
- Adjustable color palette • Easy to use
- Requires FutureSound or Perfect Sound hardware
- Amiga 500, 1000, and 2000 compatible
- Variable frequency analysis
- Documentation included • and more!

VISA/MasterCard accepted - no CODs please

Send \$29.95 plus \$3.00 shipping/handling to:

(Calif. residents add 6% sales tax)

**DIGITAL WIZARDS, INC.**

9307 Carlton Hills Blvd.

Santee, CA 92071

(619) 449-5218

Music Visions is copyrighted by Digital Wizards, 1988  
Amiga, Future Sound and Perfect Sound are registered  
trademarks of Commodore-Amiga, Inc., Applied Visions  
and SunRize Industries, respectively.

depending on the situation. The designers chose, unwisely, to lip-sync the finely detailed characters with the Amiga's croaky speech synthesizer.

While initially intriguing, these encounters ultimately become extraneous. Part of the problem is the limited alternatives available. More significantly, the characters' personalities do not vary enough to make the interrogations challenging. Bribing or threatening the noble Mohammed Bahjat al-Iiwa are equally ludicrous propositions. Conversely, the oily businessman, Mobido Kamuzu, is always ready for extra silvers. If you can psyche out the characters in the first episode, you've got this part of the game licked.

Following the questioning phase, you board the amphibious Advanced Robotic Transcoordinator (ART) for a flight to the sight of the episode's drama. ART is equipped with communications gear, a medical center, and a host of scanning devices to sweep the ocean floor in advance of your dive.

The diving sequences are fascinating, as you share the sea with jellyfish, sea-horses and other assorted fauna. You will also face adversaries as natural as strong ocean currents and as unnatural as the deadly robotic crabbots. As you encounter missions as diverse as treasure hunts, rescues and the exploration of a sunken Incan city, you can make use of scanners, transporter beams to and from ART, and of course, the mandatory cute robot, RUF (Remote Underwater Friend), who you may send on independent missions. In fact, if you don't put RUF to work, he will pout and report he no longer likes you. Just what you needed.

Though there are fourteen different episodes on the disk, you cannot jump ahead, say, from episode 2 to episode 10. One reason for this is that the episodes actually build on each other. While this does tend to keep players coming back out of curiosity, some will never see the higher numbers. Once an episode has been conquered, it may be "rerun."

Meeting success in Return to Atlantis requires patience, some insight, and a better-than-average sense of direction. The game is on two disks, making a second drive desirable, though not necessary. Mouse control is possible, but why be a masochist? Use a joystick.

After several sessions with Return to Atlantis, I can't shake the feeling that the designers' original vision got sidetracked somewhere. As good as the game is, it has the feel of an unfinished symphony. I like Return to Atlantis, but as with the lost continent itself, we may never know what happened to it.

•AC•

**ROCKET RANGER**

Cinemaware Corp.

4165 Thousand Oaks Blvd

Westlake Village, CA 91362

(805) 495-6515

\$49.95

1 player

**COSMIC RELIEF**

Datasoft, Inc.

Suggested price \$34.95

1 player

**ROAD WARS**

Arcadia

Suggested price \$34.95

1 - 2 players

**RETURN TO ATLANTIS**

Suggested price \$49.95

1 player

Marketed by Electronic Arts

1820 Gateway Drive

San Mateo, CA 94404

(800) 245-4525

**SUB BATTLE SIMULATOR**

Epyx

600 Galveston Drive P.O. Box 8020

Redwood City, CA 94063

(415) 368-0606

Suggested price \$49.95

1 player



by John Steiner

# Bug Bytes

## The Bugs & Upgrades Column

Ted Michelsen of Lakewood, CO reports a bug fix for the Epson JX-80 printer driver for IntroCad from Progressive Peripherals & Software. The problem stems from the third and fourth lines in the driver's "Final" copy segment. These lines are interchanged, and the error produces a mixed up print. If you wish to tackle the correction yourself, use TxEt, ED or any other editor. The correct data items are:

FINAL 240 216 1920 0 8 2	
12 1B 40 1B 23 1B 55 01 1B 33 01 0A 0A	(line 1)
4 1B 5A 80 07	(line 2)
9 0D 0A 1B 33 16 0A 1B 33 01	(line 3)
2 0D 0A	(line 4)
3 0C 1B 40	

As reported in an earlier "Bug Bytes," Progressive Peripherals & Software also has an upgrade available for IntroCAD. Among other things, the upgrade fixes the bugs in the printer drivers.

Will Murphy of Victoria, BC reports problems with saving graphs in IFF format in Analyze! 2.0. Members of a local users group report the same problem. Saving as a Graph Picture file caused an Amiga crash. I have saved this way myself with version 2.0 and have not had any problems. If you have any ideas about what the problem might be, please send them along. Will commented that he has written to Micro Systems Software about the problem and has not yet received a reply.

Dan Pierce reports problems with several game programs. Dan writes that Fire Power freezes up after some time in the

two-player mode. The freeze usually occurs when you are on the verge of capturing the opponent's flag.

Aegis Software's Ports of Call creates some questions unexplained in the manual. Dan would like to know how time delays are applied when loading and unloading cargo, docking, etc.

According to Dan's letter, Command Simulations' Blitzkrieg At the Ardennes visits the Guru if you try to play against the computer. If you have similar problems or fixes to the above problems, send them in.

SuperView is a picture display utility that shows virtually every kind of IFF graphic file. The program has a bug that has some people puzzled—it does not always display a file. Experimentation showed that SuperView does not display a file that contains a space in the filename. Rename a picture file to remove spaces, and SuperView displays it.

The Accountant from KFS Software is now shipping version 1.1. The modular accounting system retails for \$299.95. An additional \$24.95 signs you up for life membership in KFS Update Club. As a member, you get updates for \$9.95 per disk, a newsletter, and telephone support. Software upgrades are downloadable. Call KFS for complete upgrade details.

**KFS Software**  
1301 Seminole Blvd., Suite 117  
Largo, FL 34649  
(813) 584-2355

If you are using ProWrite from New Horizons Software with a 512K Amiga, you may have trouble getting an acceptable printout with your wide carriage (15 inch) printer. If so, contact New Horizons Software for the available bug fix. The fix is version 2.0.2, but no other fixes or upgrades from the previous version are included. If you are having problems printing from your 15 inch printer, return your program disk and request the latest version.

**New Horizons Software, Inc.**  
Box 43167  
Austin, TX 78745  
(512) 328-6650

Commodore Technical Support has a new hard disk driver utility for use with the Amiga 2000 and the A2090 hard disk controller card. To install the driver, just copy over the existing driver to the Expansion drawer. You don't need to reformat the drive or change anything else on the hard disk. Rebooting the Amiga enables the new driver. See your Amiga dealer for the upgrade program. If your dealer doesn't have it, you can find it in the download libraries on People Link and on many other information services.

Supra Corp has finished version 4.3 of Supra Boot Utilities disk. If you have a Supra hard drive, you can take advantage of these programs. The new programs take advantage of the Fast File System, soon be released with Workbench 1.3. The Supra utilities are available directly from Supra Corp, or you can find them posted to various information services. If you don't have a Supra hard disk, these files are of no value to you.

(continued)



**Supra Corp**  
1133 Commercial Way  
Albany, OR 97321  
(503) 967-9075

Shakespeare Version 1.1 from Infinity Software is now available. If you mailed in your registration card, you already have the upgrade. If not, send in your registration card to receive the latest version. Version 1.1 fixes many of the problems in the original. The upgrade package includes upgrade information and two disks—a program disk and a clip art disk with templates for business cards, letterhead, and a newsletter.

**Infinity Software**  
1144 66th St Suite C  
Emeryville, CA 94608  
(415) 420-1551

Two months ago, I reported that Digi-View 3.0 would be ready soon, though no details on ordering the upgrade were then available. Specifics on the upgrade program have now been released. If you have Digi-View 2.0, the upgrade is simple—send the original 2.0 disk with \$14.95 + 3.00 shipping and handling to New-Tek. They will send version 3.0. An upgrade is also planned for Digi-Paint to be released "in a couple of months."

**New Tek, Inc.**  
115 West Crane  
Topeka, KS 66603  
(800) 843-8934

Thanks to Steve Holaday of Idaho Falls, ID and Albert Harsch of North Huntingdon, PA for sending information and a copy of the company newsletter from OXXI, Inc. We reported on the latest Oxxi upgrades last month, but they have moved since then. The phone number listed in last month's column is correct, but the address is no longer current. The correct address is:

**Oxxi, Inc.**  
P.O. Box 90309  
Long Beach, CA 90809-0309  
or  
1339 E. 28th St.  
Long Beach, CA 90806  
(213) 427-1227

Mr. Harsch also passed along information about the bug that causes WordPerfect files to be date stamped one day earlier than the correct date during leap years. The bug, mentioned in AC V3.5, has been fixed in the latest versions. The current version of WordPerfect is dated 4/15/88. To find out which version you have, press the Help key when running WordPerfect; the version date is listed in the menu bar. Upgrades are free to registered owners; just call the customer service number and ask for details of the date bug fix. WordPerfect is also offering registered owners a copy of WordPerfect Library for \$70.00 (Library retails for \$129.95). The Library contains several utilities, including a calendar, a scientific calculator, a program editor, and a couple of simple file management tools.

**WordPerfect, Inc.**  
288 West Center St.  
Orem, UT 84057  
(800)321-4566 (order line)  
(800)321-5906 (technical support)

Progressive Peripherals & Software announced upgrades to three software packages in the June/July issue of their newsletter. As mentioned earlier, one of those programs is IntroCAD.

VisaWrite Desktop version 1.09 has been improved with increased speed, faster screen updates, and better reliability. Improved printer driver operation through preferences also provides greater performance. A free upgrade of Superbase Professional, version 2.03, is being sent to all registered users. Several bugs in the original have been corrected.

If you have questions about any of these upgrades, or you wish to register your software to take advantage of the automatic upgrades policy, contact:

**Progressive Peripherals & Software**  
464 Kalamath St  
Denver, CO 80204  
(303) 825-4144

That's all for this month. If you have any workarounds or bugs to report, or if you know of any upgrades to commercial software, notify me at:

**John Steiner**  
c/o Amazing Computing  
Box 869  
Fall River, MA 02722-0869  
...or leave EMail to  
Publisher on People Link or  
73075,1735 on CompuServe

•AC•

*[In the past few months, our "Bug Bytes" column has suffered from a few bugs of its own, as follows:*

- In V3.5, we reported an erroneous problem and solution for The Right Answers Group **Director** software. We stated that "if files have been modified by a program other than Deluxe Paint, the Director gurus on certain IFF files. The solution is to load them into Deluxe Paint and re-save."

The Right Answers Group informed us that some PixMate and Photon Paint files caused problems that resaving did not solve. Two solutions are offered. First, *DFilter.arc*, a program that corrects the incompatibilities, has been uploaded to PLink and CompuServe. Second, an upgrade is available from the Right Answers Group. If you have a version dated before 3/22/88, send \$10 with your original disk to the address printed on the disk pocket in the manual.

We also reported a problem with single-letter variables in the Director. The Right Answers Group stated that there is no problem, but users should note that variables names are case-sensitive in the Director. ("A" and "a" are not the same.)

- In V3.6, we noted a bug with the defaults in Gold Disk's **Professional Page 1.0**. What we failed to mention was that Professional Page 1.1, with the bugs corrected, had been shipping for some time. 1.1 was shipped to all registered users free of charge.

Our apologies for any inconvenience go out to The Right Answers Group, Gold Disk, and all owners of the mentioned products. If you find incorrect information anywhere in AC, please contact us in writing immediately—Ed]



# Speeding Up your System

## with Floppy Disk Caching

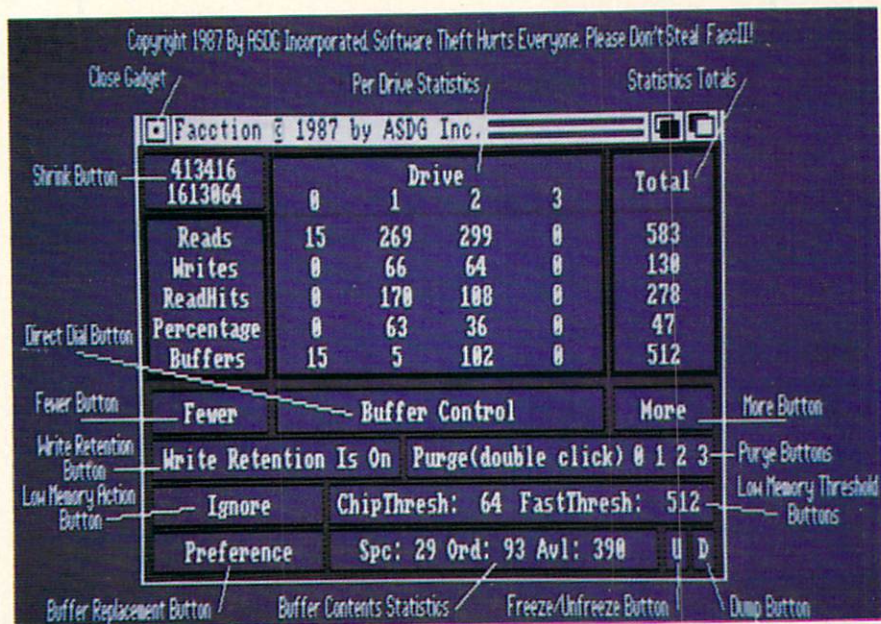
by Tony Preston

For many of us, hard disk drives are too expensive to justify their purchase. One of the reasons for adding a hard drive is to speed up the disk I/O. A much less expensive alternative anyone can afford is disk caching. If you are a system guru, feel free to skip the next few paragraphs, since you probably know more about the subject than I do. People not quite so high in studies of the mystic arts, please read on for an introduction.

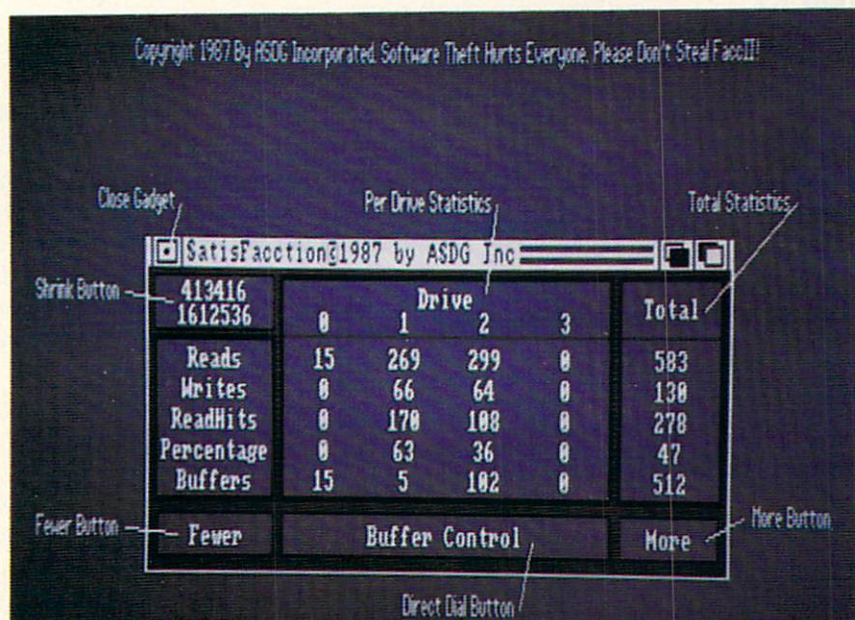
First, a couple of definitions. Cache is very fast storage, designed to increase the execution speed of running programs. Cache is generally transparent to the running programs. A disk cache is a set of buffers in memory which is filled by your normal disk I/O. Each time your program makes an I/O request, it checks if the sector needed is in memory before a real I/O is done. Higher operating speeds can be achieved by utilizing memory to "cache" disk sectors. When a sector is found in the cache, and a real I/O is eliminated, you have what is called a "hit" on the cache. The greater the number of buffers available for cache, the larger the "hit" rate. Unfortunately, doubling the number of buffers in a cache will not double the hit rate.

Our disk drives have 160 tracks with 11 sectors on each, for a total of 1760 sectors per disk. The system reserves two sectors at the start of the disk, leaving 1758 for data and directories. The file manager is the central system process which reads and writes sectors to the disk based on what you do. Any time you type in a command, open a file in a program, access a directory, or delete an old file, the request results in several I/O operations. For example, you type in the command "DELETE a.txt". The system first looks in the current directory for the command DELETE; not finding it there causes it to look for C:DELETE. If it is in the C directory, the file is read and loaded into memory.

After the file is loaded, it is executed. The DELETE command then looks at the current directory to see if "a.txt" exists. The file is



Facction by ASDG, Inc.



Satisfacction by ASDG, Inc.

(continued)



# Nothing but the best.

Now even better  
with twice the  
showroom space!

Still at...

396 Washington Street  
Wellesley, MA 02181  
(617) 237-6846



Authorized Commodore  
Amiga Dealer and  
Commodore Service Center

FIGURE ONE

```
read-rtc  
arun faccii 512 noio pri 15  
cache -b ?  
execute build_program  
cache -b ?
```

Startup-Sequence for disk performance test

FIGURE TWO

```
read-rtc  
arun faccii 512 noio pri 15  
cache -b ?  
DISKPERFA>test_output  
cache -b ?
```

Startup-Sequence for disk performance test

file managers I/O requests. The noise is generated every time the disk moves the heads during a seek—one of the slowest operations the disk drive can perform.

The Amiga trackdisk device does its I/O in tracks for faster access. Reading a track just to get one sector sounds like a waste of I/O. Most often, I/O requests read sequential blocks of data which all appear on the same track. The first read on a track does the actual I/O; the rest just copy the data from the track buffer. This works great until you have your files scattered over many different tracks (fragmentation). When a file is fragmented over the disk, the access slows and you hear the GRONK-GRONK.

To solve this problem, a cache is used to keep the most frequently accessed sectors in memory buffers which eliminate the need to do actual I/O. (It also eliminates the GRONKS.) Commodore created the ADDBUFFERS com-

mand to solve this problem. Other programs, such as FaccII and BlitzDisk, also create a disk cache and improve system performance. This article gives you some information on these programs and their use. I have performed tests on my system in an attempt to judge the different methods available for speeding up your system.

This article describes how I compared the three available disk caching methods. Each test was run from a freshly booted system set up to run the test and take the timings needed.

The first method of comparison was to use the program DISKPERFA from Fred Fish Disk 48. DISKPERFA attempts to measure the I/O speed of a drive. All results below were derived from a single run of the program from a freshly booted system. DISKPERFA was the only program running. It should be noted that DISKPERFA may not be the best program for this kind of test; it is just one of the methods I used for this comparison. See Figure 2 for the startup sequence.

The second method of comparison times a series of test compiles followed by a link (build a program from source files). The result was timed using the DATE command both before and after. See Figure 1 for the startup sequences.

In each case, I set up the disk cache program, rebooted, and executed a script which ran the test. Each compile test started with a clean disk, compiled several programs, and linked the programs into one executable. A DATE command included in the script saved the time before and after the test.

Each time DISKPERFA was run the output was placed in a file. The runtime parameters are listed with the results.

Four test situations were run for each test. Each program was set up in the startup-sequence, the system re-booted, and the test run. Since each test took a while to run, the timing was done with the DATE command. All tests were run with Kickstart version 1.2 on an A1000

deleted by updating the directory blocks involved and setting the bitmap on the disk to indicate that the sectors used by the file are available again. This command creates many disk I/O operations that are one sector in size. As you may notice, when you enter the command, you hear a GRONK-GRONK noise as the heads move around in response to the



with 3 drives and a 68010. All tests were run from CLI without the Workbench loaded. In each case, the system was set to have 512 sector buffers. The tests labeled "plain" are set up the same way as the other tests, except no caching program is running.

### The First Test:

Product	buffer size: 512
	I/O rate
AddBuffers	5637 (bytes/second)
FaccII	9395
Plain	11808
BlitzDisk	19275 ← faster rate

The results of the first test indicate that performance actually suffered for all the disk cachers except BlitzDisk. This is due to the I/O patterns used by the DISKPERFA program. It attempts to defeat caching and measure the I/O speed of the drive. This data is presented just to prove the point that any benchmark must be understood before it is quoted as the "truth." We tend to take items like this and say that one program is better than another. You must judge these results and apply them to your situation. The only program showing a gain in performance is the BlitzDisk program which proves to be about one-and-a-half times faster than a plain system (no caching).

### The Second Test

Product	Reads	Hits	Percent Hits	Time	Improvement
FaccII	3772	2440	64.7%	7:52	26 %
BlitzDisk	4251	2342	55.1%	8:15	22 %
AddBuffers	—	—	—	10:21	2 %
Plain	—	—	—	10:35	—

This result is expected. It shows that both commercial programs (FaccII and BlitzDisk) improve system performance by over 20%. The AddBuffers, while improving the system performance by 2% in this test, eats up about 256K of chip memory! Both BlitzDisk and FaccII use external (or Fast) memory if you have it conserving chip memory.

The conclusion of this test is that either commercial program is a much better choice for a user with expansion memory. The test indicates that FaccII is better in the Compile/Link test, and BlitzDisk is faster in the performance test (not a normal situation). Both programs have some definite advantages over AddBuffers, since they are faster and do not force you to use chip memory. Both programs also allocate their buffers from a common buffer pool, which doesn't force you to allocate fixed amounts for each drive. This is a definite plus for those with 3 or 4 drives!

There are some other factors to consider as we look to see which program best does the job of reducing the real I/O the system has to do. Some of these factors are:

1) *The use of Chip memory by AddBuffers limits the usefulness with programs like Deluxe Paint which need a lot of chip memory for screens.*

2) *A "smart" algorithm which gives preferences to directory sectors since they are more frequently requested.*

3) *Dynamic control over buffers in use.*

4) *Low memory detection.*

5) *Influence over what the disk cacher will keep in the cache.*

Now we come to the part where I decide which program is the best for my system. Note that I am considering my system, not yours. I can only give you the facts (flavored with my own likes and dislikes) that I used to evaluate these tools. You have to decide for yourself.

BlitzDisk is small and doesn't include a lot of user interface niceties. It does have some useful features, such as allowing you to cache your hard drive if you have one. It allows you to cache directory blocks only and to prevent the use of

(continued)



## THE HOTTEST AMIGA STORE IN THE COUNTRY.

Amazing Computer Systems is hot. Our sizzling selection of Amiga products has become the talk of the town. From more than 700 software titles in stock to the hottest selections of

hardware, accessories and books. All at red-hot prices. So remember. When you're hot, you're hot. And when you're not, you're not shopping Amazing Computer Systems.

### Amazing Computer Systems, Inc.

Frankford Village Shopping Center  
3030 N. Josey Lane #144 (2 doors South of Skaggs) Carrollton, Texas 75007 (214) 394-8383  
Open Mon.-Sat. 10am-6pm Thurs. 10am-8pm

Amazing Computer Systems is not affiliated in any way with PIM Publications, Inc. or Amazing Computing.



# C Programmers SOFTWARE INGENUITY

Presents

## The C Toolkit Source for 200+ C Functions

Including:  
Binary/Hex/Octal  
String Comparison  
Date and Time  
Math/Fractions  
Memory Manipulation  
Modulous 10/11

5 Different Methods for Sorting  
Strings and all other Data Types,  
including a Non-Recursive  
QuickSort that will sort 1000  
integers in 1.46 seconds!  
Random Number Generator  
250-page printed Manual

### Special Introductory Offer

**Only  
\$29.95**

(Suggested Retail \$39.95)

Plus \$3.00 S & H (USA)  
FL Residents Add 6% Tax

Send check or money order to:

**Software Ingenuity**

11325 94th Street North

P.O. Box 10084

Largo, FL 34643

(813) 393-8240 for info/COD

chip memory for buffers. Since I was trying to measure performance, I also noticed that BlitzDisk, while it reports the value of disk caching, does not indicate which drive the statistics are for!

BlitzDisk is a freebie included in the TxEd Plus package. It comes with about 8 pages of documentation detailing the few options you have. The manual also indicates that some hard drive backup programs have had problems when running BlitzDisk. The manual suggests you turn it off before running any special programs like DiskDoctor. While BlitzDisk seems to work fine, it is not the program I use for several reasons:

- 1) It doesn't give me enough control over things.
- 2) It is a little slower than FaccII (see second test).
- 3) I don't like having to shut it down for certain programs.

My choice is FaccII. I used both BlitzDisk and FaccII for about two months each and really found few differences except for the things mentioned here. Both run as background tasks and do their job without creating new problems. I choose FaccII based on the user interface, the control it gives me, and the fact that I do not need to shut it down when running any other program.

FaccII can be controlled from any of three different interfaces. Each interface allows you to control the number of buffers and displays some statistics on the effects of caching. Satisfaction is an older style interface, much like the display from FaccI. It doesn't give you access to several of the advanced features available in FaccII.

Faccion is a newer display that includes controls for all the advanced features. I used the Faccion interface for a while and changed to another program found in the PD world, AFacc (I call it CACHE on my system). AFacc is a PD program

written by Andry Rachmat. It is a much smaller program designed for CLI-only use. I stay in the CLI environment most of the time and find everything I need in this program.

One of the reasons I chose FaccII is the programmer's interface. FaccII allows any program to adjust the cache parameters. FaccII comes with two manuals on the disk (which is fortunate—my bookshelves have enough manuals already). Each manual gives extensive details on the operation and use of the program. I wish every program came with so much documentation! I found speed-ups with all kinds of processing with FaccII. I even installed it in the startup-sequence of some of the games I own. It makes disk access faster whenever it is used! I like the FaccII features better than BlitzDisk features. While both work quietly in the background, I didn't worry about whether I needed to shut down the caching with FaccII. The lack of worry, speed improvement, decrease in GRONKS, and the nice interface make me a FaccII user all the way.

One very nice feature included with FaccII is the upgrade policy. I called ASDG for information about FaccII and found there are plans for a FaccIII which will have some very nice improvements. (I was asked not to spill the beans.) ASDG has a policy which says "ASDG Inc. fully intends to release future upgrades to FaccII." As a FaccII owner, you are entitled to any and all future upgrades to this product with return of your original disk, a self-addressed stamped envelope, and a small handling fee." I think every software producer should follow this fine example! It is nice to know I can upgrade without paying the full price again!

•AC•

*FaccII is a registered trademark of ASDG, Inc.  
BlitzDisk is a registered trademark of  
Microsmiths Inc. AddBuffers is an AmigaDOS  
command supplied by Commodore with V1.2*





1&gt; by Rich Falconburg

# The COMMAND LINE

## *The Continuing Guide to the CLI*

As we have already seen, information may be easily organized using subdirectories. On a storage device of any size, path names often become quite long and cumbersome. It's not uncommon to encounter program file requesters that will not allow entry of the full path. What's the use of organizing your files so neatly when you can't get to them?

Fortunately, AmigaDOS provides an elegant solution to this problem: the ASSIGN command. Some time ago, I had difficulty entering this path in a certain paint program (too many characters):

```
GRAPHICS:DPAINT/PICTURES/HIRES/BW/Label.pic
```

To overcome this problem, I put the ASSIGN command to work. ASSIGN establishes a connection between a name you provide, referred to as a "logical name," and a volume, directory, or command the system uses. AmigaDOS then translates the logical name as the equivalence of the string given in the assignment. For example, to shorten the above path, I established the logical name "BWPICS:" with:

```
ASSIGN BWPICS: GRAPHICS:DPAINT/
    PICTURES/HIRES/BW
```

The name you provide immediately follows the ASSIGN command and must always end with a colon. We have encountered the colon (:) before when discussing volume names. All volume names are logical names and therefore establish an automatic assignment to the ROOT directory of a given disk. The path or command the assignment is made to follows the logical name. We may use

the logical name as if it were a device. For example, I may now enter

```
1> CD BWPICS:
```

and my current directory will be set to

```
GRAPHICS:DPAINT/PICTURES/HIRES/BW
```

This opens a new avenue in directory navigation. You can place assignments to your favorite directories in the startup-sequence. A word of caution here: ASSIGN expects the given path to exist. If you make an assignment to a volume and directory not currently mounted, you get a requester prompting you to mount the appropriate volume.

Logical names are beneficial in that they help avoid device dependence. For example, the standard logical assignment SYS: is another name for the system disk. Several other logical assignments are performed by AmigaDOS for directories it needs to operate. These can be displayed by entering the following:

```
1> ASSIGN      or
1> ASSIGN LIST
```

```
Volumes:
Workbench [Mounted]
```

```
Directories:
SYS      Workbench:
S        Workbench:S
L        Workbench:L
C        Workbench:C
FONTS    Workbench:fonts
DEVS     Workbench:devs
LIBS     Workbench:libs
```

```
Devices:
DF0 DF1 PRT SER PAR
RAW CON RAM
```

You should get a listing similar to this one. If a volume has assignments to it and is not currently mounted, it will show up in the "Volumes:" list without the "[Mounted]" designator next to it. The "Directories:" list shows each logical name on the left with its equivalent on the right. Assignments you make also show up here.

To use a different disk as the system (Workbench) disk, it's necessary to change the assignments of the logical names shown. I have a partition (volume) on my hard drive labeled WORKBENCH that I use as my system disk. I first change the assignment of SYS: to point to a new volume with

```
ASSIGN SYS: DH0:
```

I then reassign the rest of the needed directories.

```
Assign C: SYS:C
Assign S: SYS:S
Assign L: SYS:L
Assign Fonts: SYS:FONTS
Assign Devs: SYS:DEVS
Assign Libs: SYS:LIBS
```

You can use this same procedure to reassign Workbench to any volume. You may have noticed that I'm including a logical name as part of the assignment to each directory. An ASSIGN LIST shows that the equivalence string points to the original assignment. That is, SYS: is equal to WORKBENCH:, thus C: is equal to WORKBENCH:C, S: is equal to WORKBENCH:S, and so on. The advantage of this approach is that you only need change the assignment to SYS: to use a different device. If the rest of the



# NOTICE!

## Amiga CLI, part 1 Update!

If you responded to our Reader Service Card and are anxiously awaiting your copy of *Amiga CLI, part 1*, don't dismay. The release of the first CLI booklet is pending the official release of Workbench 1.3. As soon as Commodore releases 1.3, *Amiga CLI, part 1* will be sent out to all who responded. Thank you for your patience and your support of the Amiga.

assigns are in a script file, and you hard code the device in each of the ASSIGN statements, you must edit the file and change each occurrence of the device name to the new one. With the approach shown above, I can enter:

```
1> ASSIGN SYS: DF0:
1> EXECUTE Assigns.stu
```

The file "Assigns.stu" contains the assignments shown above. I do not have to edit anything, and I can change it back just as easily. You can create batch files with assignments needed by some programs and then mount the program floppy and EXECUTE that batch file. To DE-ASSIGN a logical name, enter it without the equivalence string.

```
1> ASSIGN SYS:
```

### Who Goes There?

Since the device DH0: is not among the standard ones supplied with the system, it's necessary to inform AmigaDOS about the device and how it is to be used.

When we took the Workbench tour, I mentioned that the devs: directory contains files called device drivers. Logically, for the system to use a device, it must have access to a driver. However, this is not enough to use the device in the system. To make a new device visible to the system, it's necessary to use the MOUNT command. In turn, the MOUNT command must know what to tell AmigaDOS, which it does by reading the MOUNTLIST file located in the devs: directory. The list is a simple text file containing specific information for each device. Here's a sample MOUNTLIST file.

```
/* ===== 10 MEG Hard Disk Drive MountList ===== */
```

```
/*          WORKBENCH          */
DH0: Device = SCSI.device
Unit = 1
Flags = 17424438
Surfaces = 4
BlocksPerTrack = 26
Reserved = 2
Interleave = 0
LowCyl = 0 ; HighCyl = 100
Buffers = 9
BufMemType = 0
#
```

```
/* Mount a 5.25" disk drive to be mounted as DF2: */
```

```
DF2: Device = trackdisk.device
Unit = 2
Flags = 1
Surfaces = 2
BlocksPerTrack = 11
Reserved = 2
Interleave = 0
LowCyl = 0 ; HighCyl = 39
Buffers = 20
BufMemType = 3
#
```

The MOUNT command ignores comments between the /\* and \*/. Each entry begins with the logical device name and must end with the pound sign (#). Most of the information contained in each entry is provided by the manufacturer of the new device. In this case, the hard drive described (DH0:) is connected to a C Ltd. SCSI controller. The value for each entry is determined with the aid of utility programs delivered with the controller. Let's look at a more familiar device and examine the most significant lines.

The entry shown for DF2: is the one from the MOUNTLIST included on the original Workbench disk. Attentive readers will have noticed that the device driver listed is not in the devs: directory. This driver is one of several that are part of the ROM Kernel of the operating system. Several drivers are essential to the Amiga's operation and are permanent parts of the system. The "trackdisk.device," "keyboard.device," "console.device," and others are always in memory. When you mount a new device, the driver described in the MOUNTLIST entry is copied to memory (if it does not already exist there).

"Unit" is the physical device address. Each device must have a unique address for the system to talk to. The unit number is a physical address and is part of the hardware device controller. The "Flags" entry is used to set any of 32 certain conditions in the controller or device. The "Surfaces" entry describes the number of sides accessible on the disk. Each surface normally has a Read/Write head associated with it. Floppy disks have only two surfaces; a hard disk may have as few as two or as many as thirty or more surfaces. "BlocksPerTrack"



is the number of 512 byte blocks on each track of a Surface, and the "LowCyl" and "HighCyl" numbers indicate the beginning and ending track of every surface. (A track is each concentric circle formatted on each surface.) A cylinder is all of the tracks of each surface for a given track/cylinder number.

The "Interleave" factor is normally determined during formatting. To oversimplify, interleave is the number of blocks skipped on a single pass. For example, if the Interleave is set to 2, every other block is read or written for each pass of the Read/Write head. This is used to improve the rate of data transfer. If the system is too slow to read or write each block consecutively on a single pass, it must wait for the block to spin around and then pick it up on the next pass. This wait period increases the time needed to transfer the data. Interleave helps alleviate this by keeping a steady flow of data coming from the disk drive.

Another way to help smooth out the data flow is to set up a type of reservoir for the data. This is done with something called "Buffers." The number given here establishes a pool of memory (512 bytes for each buffer) used to store up the bytes of data as they are moved to and from the device. This is useful because even the fastest disk drive is slower than memory and the CPU. The central processor is capable of munching data faster than a storage device can deliver it. Memory, on the other hand, works at or near CPU speed. A buffer acts like a dam, holding the data in a large collection area, and is able to dump it into the stream going to the device or CPU at the top speed it is capable of processing the information. This allows the CPU to spend more time working than waiting. The "BufMemType" value indicates the type of buffering used.

The ADDBUFFERS command allows you to increase the number of buffers for each device including DF0: and DF1:. To use it, you supply the device name and the number of 512 byte buffers to add.

```
1> ADDBUFFERS DF0:32
```

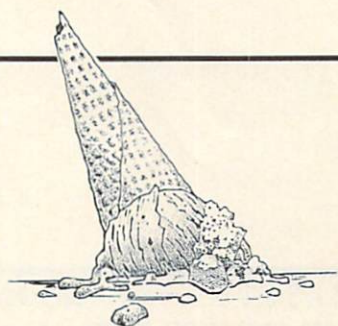
This statement adds enough buffers to hold an additional 16K of data. This memory is taken from the available system memory and not returned without re-booting. The addition of buffers to a floppy drive can significantly increase the speed of some disk operations. You have to experiment to determine the best numbers to use.

Each device to be mounted must have an entry in the MOUNTLIST file. The MOUNT command is used to inform the system that a new device exists. The only parameter required is the name of the device as given in the MOUNTLIST file, which may be anything you call it.

```
MOUNT DH0:
```

```
MOUNT DF2:
```

This brings up a small conflict of terms. A "mounted" volume is a storage instrument placed within a physical device and is available for immediate access. A "mounted" device is a unit that is part of the computer system. A



## Oops... Corrections

*It's time again for the editors at AC to blush and confess our errors. Please check out our most recent slip-ups.*

*Our "A First Look at Interchange" (AC V3.7, p. 15) mentioned polygons that aren't rendered correctly when translated by Interchange from VideoScape 3D and Sculpt 3D. We failed to note, though, that the problem does not stem from Syndesis's Interchange. Rather, the rendering quirk can be traced to VideoScape 3D and Sculpt 3D. The manufacturers have been notified and a follow-up review will appear soon in AC.*

*Also, the Software Visions ad on page 13 of AC V3.8 listed an incorrect telephone number. Please note the correct number for Software Visions: (508) 875-1238. As always, Software Visions toll-free (800) 527-7014 is also available.*

*Our apologies to Syndesis, Software Visions, and all our readers. If you find any errors anywhere in AC, please contact us immediately in writing.*

mounted device may or may not provide storage capability. For example, CON: is a mounted device that provides keyboard input to, and video display from the computer's operating system. A device may only be mounted once. A volume may be mounted and dismounted many times. This is normally transparent because of certain built-in features of the Amiga's floppy controller. On the other hand, if you own a 5.25" disk drive (or other removable media device), you must use the DISKCHANGE command to inform the system that a new volume has been placed in the drive. The DISKCHANGE command dismounts the old volume and mounts the new one. To use the DISKCHANGE command, supply the device name:

```
1> DISKCHANGE DF2:
```

This causes AmigaDOS to mount the new volume. Contrary to popular belief, this command may be used with any disk device. One very useful application is with removable hard disks.



```

A M A S D I S-----inx#-day#-ldzx-stzx-beq^+add#
A RISC DEVELOPMENT SYSTEM...Assembler-Disassembler-Linker
For ALL Amiga using 4 Char. 6502 "like" MNEMONICS:
Subset of 50 Generic 68000 instructions, for FAST, Coding.
RELOCATABLE OBJ MODULES, LINKED to THEIR OWN ZP RAM AT
RUN TIME. [32 kbytes Max.] Source From ED or AMASDIS,
Disassem. Printout, ON GCN HELP. AmigaBasic Param's passed
Thru ZP.TEXTFILE GUIDE. Port to ROMS for SBC FFAST DEVEL.
X reg INDEXED INDIRECT[16bit offset to 24 bit Base Add's]
THE PERFECT DEVELOPMENT SYSTEM for REAL-TIME APPLICATIONS.
ADD UP TO 50 ADDITIONAL INSTRUCTIONS....[YOUR MNEMONICS]
Asmdis/RuntmDis/ObInk-----A M A S D I S from AELEN ELECS
$39.95 407 W. 40th
[Amiga/AmigaBasic are COMMODORE Products] NY NY 10018

```

The MOUNT command only connects the device to the system; it does not cause the device to appear on the Workbench or in the list given by INFO. The DISKCHANGE command is useful for making the device visible to the user.

To get information about the devices and volumes currently in the system, use the INFO command.

```

1> INFO
Mounted disks:
Unit Size Used Free Full Errs Status Name
DH1: 5.3M 7466 3244 69% 0 Read/Write PUBLISH
DH0: 5.2M 10180 322 96% 0 Read/Write WRKBCH
DF1: No disk present
DF0: 880K 1748 10 99% 0 Read Only WB_1.2
RAM: 19K 36 0 100% 0 Read/Write RAM DISK

Volumes available:
RAM DISK [Mounted]
PUBLISH [Mounted]
WRKBCH [Mounted]
WB_1.2 [Mounted]
PD_Vol1

```

INFO requires no parameters and provides statistics about each storage device in the system. A word to the wise about the size value: the value shown indicates the unformatted amount of space on the drive. This is somewhat

misleading. For example, the actual usable space on DF0: is about 837K.

The numbers in the "Used" and "Free" columns indicate the number of 512 byte BLOCKS for each. The "Full" column is more useful for determining the space available. If the requester proclaims that "Volume Workbench has a Read/Write error" the "Errs" column increments the number for each one occurring on that volume. If the "Status" column says Read Only, the volume is write protected. The "Name" is the volume name, and the "Unit" is the device that volume is currently mounted in. If a volume is allocated (such as CD set to it or a file still open) and not mounted, the "Volumes available:" list shows the volume name without "[Mounted]" next to it.

#### What's in a Name?

I've talked about volume names for quite a while now without going into detail about how the name gets there. You may already be familiar with the most common route. It is quite easy to

initialize (format) a disk from the Workbench pull-down menu. The procedure may also be performed from the CLI. The formatting process establishes blocks and tracks on each surface which creates a sort of "road map" for the disk operating system to use for storing the data. The Workbench method labels the newly-formatted disk "Empty" and create a trashcan directory and icon. We have more control over this from the CLI. Here is the syntax to format a disk in DF0: with the name "My\_Disk" without a trashcan directory:

```

1> SYS:SYSTEM/FORMAT DRIVE DF0:
NAME My_Disk NOICONS

```

Insert the disk to be initialized in drive DF0: and press RETURN.

You may use spaces in the name (a real pain) if it's enclosed in quotes. The DRIVE and NAME keywords are required. As shown, the FORMAT command is found in the SYSTEM/ directory. If your startup-sequence adds a PATH for that directory, you don't need to specify it in the command line. FORMAT also works on hard disks.

What if you want to change the name of a disk after it has been formatted and now contains data? Use the RELABEL command.

```

1> RELABEL DRIVE DF0: NAME PD1
or
1> RELABEL DF0: PD1

```

The disk contents remain intact. To make a newly formatted disk "bootable," use the INSTALL command.

```

1> INSTALL DF1:

```

This writes an AmigaDOS boot block to the disk and may be performed on any floppy. It does not harm the contents of the disk.

Next time, we'll look at process control and batch operations. I'll also cover how to create and use new CLI windows.



EDUCATION EDITION

# Amazing COMPUTING™

*Your Original AMIGA™ Monthly Resource*

## AMIGA PRODUCT GUIDE

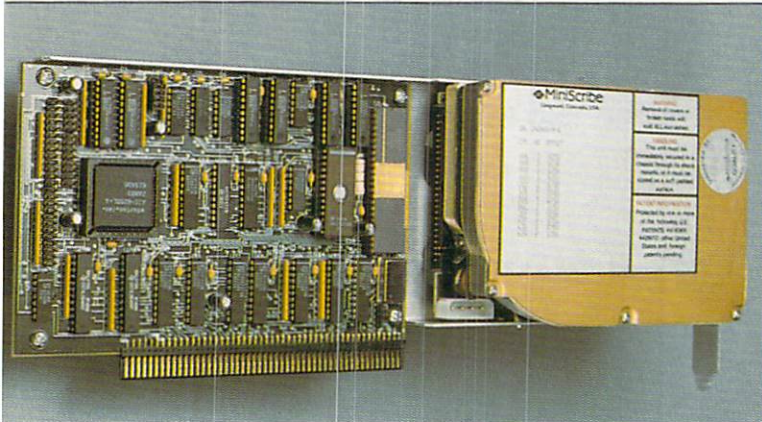
*Grammar  
Pre-school  
Educator's Tools  
Spelling & Vocabulary  
Elementary & Advanced Math  
Elementary & Advanced Reading*





## HardFrame/2000

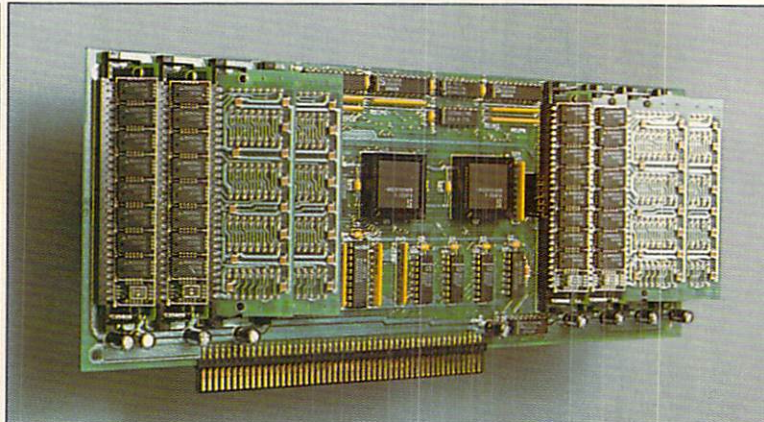
The Super-speed, DMA, SCSI Hard Disk Interface with 1.3 Autobooting



How fast is fast? **HardFrame/2000** transfers data at Amiga bus speeds! It's actually faster than the hard disk mechanism itself! And even more important in the Amiga's multitasking environment, **HardFrame/2000** has extremely efficient DMA circuitry to get on and off the bus in almost no time at all: 280ns to get on; 200ns to get off. **HardFrame/2000** autoboots under AmigaDOS™ 1.3 and is fully compatible with the new Fast File System. The core of any DMA SCSI interface is its SCSI protocol chip and DMA chip. MicroBotics has chosen the new, high performance Adaptec AIC-6250 SCSI chip, capable of up to 5 megabytes per second raw transfer speed, and the Signetics 68430 DMA chip running at 12.5 megahertz. Then we added additional FIFO buffering and enabled 16-bit wide data transfers for maximum throughput. The sophisticated design of **HardFrame/2000** provides for automatic SCSI arbitration, selection and reselection. The hardware supports either synchronous or asynchronous data transfer. **HardFrame/2000** can function as either the SCSI bus initiator or the target and can reside in a multiple master environment. Physically, **HardFrame/2000** is optimally flexible: the compact, half-size card comes attached to a full length, plated aluminum frame. The frame has mounting holes positioned to accept standard, 3.5" SCSI hard disk units such as those manufactured by MiniScribe, Seagate, Rodime, and others (hard disk mechanisms must be supplied by the user or his dealer as a separate purchase item). Alternatively, you can cable-connect to a SCSI drive mounted in your Amiga's disk bay or in an external chassis. As many as seven hard disks may be connected to a single **HardFrame**. There is no size limit on each disk. **HardFrame/2000** includes a 50-pin SCSI cable and header connectors for either 50-pin or 25-pin cable connection. Also included is a current tap to power frame-mounted drives directly from the slot itself. **HardFrame/2000** comes complete with driver, installation, and diagnostic software. Available September 1988. Suggested list price, \$329 (hard disk not included).

The **HardFrame/2000** photo shows the product with a MiniScribe 20 megabyte hard disk installed. Hard disks are not included in the purchase price of **HardFrame**. Note that if placed in the first slot, **HardFrame** uses only one slot.

## 8-UP! The Eight Megabyte Memory Card with Amiga™-specific DRAM Controller Logic



All the memory space you and your Amiga 2000 need—in a modern, highly integrated FastRAM expansion board. In **8-UP!**, MicroBotics went all the way to provide you with a truly Amiga-specific memory design to meet the special demands of the Amiga's high speed multitasking environment: The heart of any memory expansion is its *DRAM controller circuitry*. Rather than compromising with off-the-shelf parts, MicroBotics developed its own, custom controller design and built it into high-speed, *Programmable Macro Logic* chips (Signetics PLHS501). These new, super chips (each **8-UP!** uses two PML's) permit MicroBotics to employ *sparse refresh* technology to assure that your **8-UP!** is a truly zero wait-state/minimal-refresh-collision memory design. If you're putting eight megabytes in only one slot, that means that you probably have plans for your other A2000 slots. **8-UP!** gives you new freedom to do that planning since, unlike other ram peripherals, it is an extremely low-power memory card—a single, fully-loaded, 8-megabyte **8-UP!** draws an astoundingly efficient 0.800 milliamps! That's less than *two-fifths* of the power "budget" for a single slot! Low power draw also means that the card is cool-running for reliability and long life (not to mention a cooler Amiga!). **8-UP!** offers you maximum flexibility in memory configuration: it is organized into two separate *PIC's* (Amiga-speak for autoconfiguring peripherals). Each **8-UP!** PIC consists of four SIMM module sockets; these sockets accept either 256k-byte or 1 megabyte *SIMM's* (Single Inline Memory Modules). You can also purchase optional *PopSIMM* boards from MicroBotics; fill them with conventional RAM; then use *PopSIMM's* to fill your **8-UP!** The card can run with as little as 512k of memory or as much as eight megs—with many intermediate configurations possible (particularly the six megabyte configuration, most desirable for use with a *BridgeCard™*). **8-UP!** is speedy, efficient, custom memory technology for your Amiga 2000—and it's available now! **8-UP!** suggested list price is \$199 (0k installed). Optional *PopSIMM's* are \$49.95 per pair.

The **8-UP!** photo shows the card half populated with conventional SIMM modules and half with MicroBotics *PopSIMM's*. *PopSIMM's* (without DRAM installed) are available as separate purchase items.



### MicroBotics, Inc.

Great Products Since the Amiga Was Born!

811 Alpha Drive, Suite 335, Richardson, Texas 75081 (214) 437-5330

SOLD ONLY THROUGH YOUR AMIGA DEALER

Tell your dealer he can quick-order from MicroBotics directly—no minimum quantity—show him this ad!

\*Amiga is a registered trademark of Commodore-Amiga. "HardFrame/2000", "8-UP!", "PopSimm", are trade names of MicroBotics



# Amiga Product Guide

## Education Edition

<b>Educator's Tools</b>	<b>51</b>	<b>Miscellaneous</b>	<b>55</b>
<b>Grammar</b>	<b>51</b>	<b>Pre-school</b>	<b>59</b>
<b>History</b>	<b>52</b>	<b>Reading-Advanced</b>	<b>59</b>
<b>Language Instruction</b>	<b>52</b>	<b>Reading-Elementary</b>	<b>61</b>
<b>Math-Advanced</b>	<b>53</b>	<b>Science</b>	<b>62</b>
<b>Math-Elementary</b>	<b>54</b>	<b>Spelling/Vocabulary</b>	<b>62</b>
		<b>Social Studies/ Geography</b>	<b>63</b>

Amazing Computing™ provides this guide as a compilation of products from all companies who have responded to our multiple requests for information. The descriptions have been adapted directly from the information supplied by developers. Accordingly, the listings are not intended as reviews, but rather as information for active users in the ever-growing Amiga market. AC™ will not be held responsible for omissions or errors (including prices). If you detect any errors or omissions, please let us know, in writing, as soon as possible. —Ed.

### Educator's Tools

**Grade Manager** \$89.95  
An electronic gradebook. Holds information on up to 100 students, automatic conversion to 100% point scale, and more.  
*Associated Computer Services, 1306 E. Sunshine, Springfield, MO 65804, (417) 887-7373*

**Magellan** \$195  
Artificial intelligence/expert system building tool. Implements rule-based user interface that allows knowledge bases to be built using simple IF-THEN logic.  
*Emerald Intelligence, 334 South State St., Ann Arbor, MI 48104, (313) 663-8757*

**Quiz Master** \$79.95  
Allows teachers to create their own lessons in any subject. Different question formats. Supports sound, music, and graphics.  
*Associated Computer Services, 1306 E. Sunshine, Springfield, MO 65804, (417) 887-7373*

### Grammar

**Complete Practical Composition Series** \$229.95  
Practical Composition I-V. Improve your writing skills with this five-disk set.  
*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

**CP-941 Capitalization** \$29.95  
Teaches students how to capitalize words correctly within sentences.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

**Ghostly Grammar** \$49.95  
Learn grammar through tutorial and games. Parts of speech, punctuation, and sentences. Grades 3 through 12.  
*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV 89121, (702) 737-8862*

**GR-910 Basic Grammar** \$29.95  
Teaches students to identify nouns, verbs, adjectives, prepositions, conjunctions, and interjections.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

(continued)



## Multi-Forth

### The Language of Innovation

If you haven't tried Multi-Forth™ you may not have yet unleashed the full power of your Amiga. This comprehensive development environment includes:

- Local Multi-Tasking
- Built-in Assembler
- Turnkey Compiler (royalty free)
- Sound Drivers
- AmigaDos & Intuition Support
- Complete Set of Include Files
- CompuServe Public Forum
- 500 Page Manual
- New price - \$99

Call for a technical data sheet or check out our online services on CompuServe at GO FORTH.

#### Creative Solutions, Inc.

4701 Randolph Rd. Ste. 12  
Rockville, MD 20852

301-984-0262

1-800-FORTH-OK (367-8465)

Major credit cards accepted

## 68000 DISASSEMBLY

### Announcing . . .

## DSM VERSION 1.0D

DSM is a full-featured disassembler for the Amiga. Check out these features and you'll see why programmers agree, "DSM is the best disassembler currently available for the Amiga. Bar none."

- DSM disassembles virtually any executable Amiga program.\*
- DSM produces output which is 100% compatible with the Amiga assembler, *assem*.
- DSM is interactive.

\* Only programs which make use of overlays can't be disassembled with DSM.

And here's the best reason yet to order DSM.

- DSM comes with a 30-day money back guarantee. If you're not 100% satisfied with DSM, simply return the product to us for a complete refund.

To order, send check or money order to:

OTG Software  
200 West 7th Street Suite 618  
Fort Worth, TX 76102

Texas residents add  
7.25% sales tax.

Price \$67.50

### Practical Composition I-V

I, III-V, \$44.95; II, \$74.95

I—Making Words Work; II—Logical, Clear Sentences; III—Selecting the Best Approach; IV—Making Sentences Work; V—Using Words Correctly.

*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

### PU-940 Punctuation \$29.95

Teaches students how to punctuate properly.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### Sentence Completion \$35.95

Reading comprehension exercises help students improve scores on sentence completion items on college aptitude tests.

*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

## History

### American History

#### Adventure \$59.95

Students meet and identify American historical characters in a challenging game format.

*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

### History Expansion Disk for Discovery \$19.95

Grades 9 through 12. Ancient through modern history. Requires Discovery program disk.

*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

### How a Bill Becomes a Law \$49.95

Realistic interactive game in which students become Congressmen trying to pass a bill.

*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

### SS-951 Lewis and Clark Expedition \$89.95

5 disks. 50 digitized photographs and drawings provide pictorial settings for Lewis' and Clark's achievements.

*MicroEd, P.O. Box 24750 Edina, MN 55424, (612) 929-2242*

### SS-952 Across the Plains \$59.95

2 disks. 20 digitized pictures depicting the journey of Americans across the Great Plains to the West Coast.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### SS-953 Early Great Lakes Fur Trade \$79.95

3 disks and a book. Interactive text, 30 digitized pictures of the European-Indian relationship in the 17th-19th centuries.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### SS-954 Making our Constitution \$79.95

4 disks. Features digitized pictures of the people and events associated with the making of the Constitution.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### SS-955 Transcontinental Railroad \$39.95

Interactive instructional program with digitized pictures of the story of the transcontinental railroad.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

## Language Instruction

### French Grammar I, II, III \$34.95 each

Ten lesson modules including at least one review of French grammar and vocabulary.

*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610; (800) 232-2224, (203) 335-0908*

### LA-981 & 982 Learning English as a Second Language \$89.95 each

5 disks in each package. Simple English for the beginner. Features many digitized pictures and the Amiga voice.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*



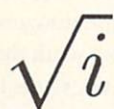
# Producing Beautiful Documents?

If you are trying to create high-quality books, technical reports, or other documents with your Amiga, you should be familiar with AmigaTeX. Nothing else will provide the power, the beauty, and the speed you want. Nothing.

AmigaTeX is not for everyone. As a matter of fact, it takes a while to master. Typesetting is not a simple art, and it calls for powerful tools. If you need interactive WYSIWYG editing, buy something else. But if you want to typeset mathematics, tables, or long documents, or simply insist on the best quality available, check out AmigaTeX.

For information about AmigaTeX, go down to your local college bookstore and look at a copy of *The TeXbook* by Don Knuth, and *LaTeX: A Document Preparation System* by Leslie Lamport. These are the handbooks for AmigaTeX; since they were typeset in TeX, they also provide an example of the capabilities of AmigaTeX. Then, write to the address below for your free demo disk.

AmigaTeX, including preview, LaTeX, and more: \$200. Printer drivers: \$100 each. AmigaMETAFONT: \$75. Experience the power. Write today.



## Radical Eye Software

Box 2081 Stanford, CA 94309 (415) 326-5312

### Language Expansion Disk

**for Discovery** \$19.95

Speech, grammar, vocabulary, communications and much more. Prepare for SATs. Requires Discovery program disk.  
*Microillusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

### Linkword Languages

\$29.95

Software and audio tapes. French, German, Spanish, Italian, Dutch, Greek, Portuguese, and Russian.  
*Artworx, 1844 Penfield Rd., Penfield, NY 14526, (716) 385-6120; (800) 828-6573*

**Senor Tutor** \$69.95

Teaches basic conversational Spanish with self-paced, interactive lessons, and digitized sound. 6 disks, pictures, dictionary.  
*Finally Technologies, 25 Van Ness Ave., San Francisco, CA 94102, (415) 564-5903; FAX (415) 626-4455*

### Spanish Grammar I, II, III

\$34.95 each

Ten lesson modules, including at least one review of Spanish grammar and vocabulary.  
*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

### Math-Advanced

#### Algebra I

\$49.95

Stand-alone, memory-driven. Evaluate, plot, simplify algebraic expressions. Work with functions, numeric expressions, geometric measurement, equations, radicals.  
*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755; (800) TR-BASIC; in NH, (603) 643-3882*

#### Algebra II

\$49.95

Stand-alone memory-driven. Systems of linear equations, inequalities, root finding, graphing, and more.  
*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755, (800) TR-BASIC; in NH, (603) 643-3882*

#### Arithmetic

\$49.95

Stand-alone, memory-driven. Evaluates numeric expressions, computes with fractions, percentages, square roots.  
*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755, (800) TR-BASIC; in NH, (603) 643-3882*

#### Calculus

\$49.95

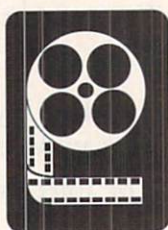
Stand-alone, memory-driven. Performs symbolic differentiation on any function you enter. Special topics include limits, curve tangents, minima, maxima, and more.  
*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755, (800) TR-BASIC; in NH, (603) 643-3882*

(continued)



# Amiga Video and Publishing

Complete line of Amiga hardware and software with over 1,200 published titles.



- ✓ Slide Production
- ✓ System Sales
  - Desktop Publishing
  - 2D/3D Video Production
  - CAD Design/Engineering
- ✓ 24-hour Bulletin Board
- ✓ Authorized Amiga Service Center
- ✓ Linotronic 300 PostscriptTypesetting

AMIGA™  
COMMODORE

Will ship to all 50 states!  
24-hour turnaround on in-stock items!  
MasterCard/Visa/AmEx phone orders!

**METROPOLITAN**  
Computer Products  
800 E. Arapaho Rd. #110  
Richardson, TX 75081  
(214) 437-9119

## Discrete Mathematics \$49.95

Stand-alone, memory-driven. Introduces computer science. Construct truth tables and Venn diagrams. Simple combinatorics and reserve functions. Graph theory, binary, more.

*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755, (800) TR-BASIC; in NH, (603) 643-3882*

## Doug's Math Aquarium \$89.95

Graphic program plots mathematical expressions in 2-D and 3-D color. *Seven Seas Software, P.O. Box 41, Port Townsend, WA 98360, (206) 385-3771*

## Math-Amation \$99.95

On-screen representation of algebra, geometry, trigonometry, calculus, physics. Fast, simple graphics, on-screen calculators.

*Progressive Peripherals & Software, 464 Kalamath St. Denver, CO 80204 (303) 825-4144; FAX (303) 893-6938*

## Math Concepts

### Expansion Disk \$19.95

Grades 9 through 12. Prepare for SATs, practice problems in algebra, geometry, trig and more. Requires Discovery program disk.

*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715*

## Pre-Calculus \$49.95

Stand-alone, memory-driven. Emphasizes how functions look, and the effect when parameters are manipulated. Explore conic sections, polar inverse functions, more.

*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755, (800) TR-BASIC; in NH, (603) 643-3882*

## Trigonometry \$49.95

Stand-alone, memory-driven. Plot your own functions and compare with tables of values. Solve triangles, build composite functions, work with trigonometric identities, more.

*True BASIC, Inc., 39 S. Main St., Hanover, NH 03755; (800) TR-BASIC; in NH, (603) 643-3882*

## Math-Elementary

### Decimal Dungeon \$49.95

Escape the dungeon master by answering math decimal problems. Incorrect answers lead to full-screen explanations. Age 9 and up.

*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV, 89121, (702) 737-8862*

### Fraction Action \$49.95

Escape the mad professor by answering math fraction problems. Incorrect answers lead to full-screen explanations. Age 9 and up.

*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV, 89121, (702) 737-8862*

### Kinderama \$49.95

5 talking games with animated graphics. Beginning addition, subtraction, upper and lower case letters, more.

*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV, 89121, (702) 737-8862*

### MA-902 Early Math \$49.95

Adding With Objects, Subtracting With Objects, What Number is Missing?, and Count 'Em.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*



**MA-904 Sum-It Mountain** \$29.95  
 Grades 1-8. Success depends on the student's ability to add numbers rapidly. Any mistake means returning to the bottom.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

**MA-908 Medal Winner** \$29.95  
 Grades 4-8. Create a multiplication problem with three or four digits that will produce the highest product.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

**MA-909 Pinball I.Q.** \$29.95  
 Grades 3-8. Division lessons combined with a video pinball game.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

**Math 1 Expansion Disk for Discovery** \$19.95  
 Basic four-function math drills for grades 1 through 7. Requires Discovery program disk.  
*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

**Math-A-Magician** \$39.99  
 Add, subtract, multiply or divide fractions or whole numbers. Play, practice and learn. Features graphics and speech.  
*The Other Guys, 55 N. Main, Suite 301D, Logan, UT 84321, (801) 753-7620*

**Math Wizard** \$49.95  
 4 educational math games for 1 or 2 players, ages 5-13. Talking program with several skill levels.  
*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV 89121, (702) 737-8862*

## Miscellaneous

**IntelliType** \$49.95  
 Sophisticated typing program combines customized lessons with an adventure story. Monitors errors, prescribes remedial drills.  
*Electronic Arts, 1820 Gateway Dr., San Mateo, CA 94404, (800) 245-8525; in CA, (800) 562-1112*

**Mavis Beacon Teaches Typing** \$44.95  
 Tailors the perfect typing course to meet individual needs. Includes explanations, typing games, and much more.  
*Software Toolworks, 1 Toolworks Plaza, 13557 Ventura Boulevard, Sherman Oaks, CA 91423, (818) 907-6789*

**Nutri-FAX** \$139.95  
 Provides nutritional data on 13 common nutrients for your own favorite recipes. Includes approximately 100 recipes for your own use.  
*Meggido Enterprises, P.O. Box 3020-191, Riverside, CA 92519, (714) 683-5666*

**SC-936 Aids Information Game** \$39.95  
 Provides basic information on the AIDS virus in a game format for 1-4 people.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

**Trivia 1 Expansion Disk for Discovery** \$19.95  
 Trivia categories include Nobel Prize winners, sports, and famous quotes. Discovery program disk required.  
*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

**Trivia 2 Expansion Disk for Discovery** \$19.95  
 Trivia categories include movies, space travel, and famous people. Discovery program disk required.  
*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

## JUMPDISK: \$5 The Original Disk Magazine for the Amiga\*

Try our new SAMPLER. It costs \$5. That's all. If you don't like it, we'll buy it back. We're that confident.

You'll get original material:

- A talking slideshow program
- A text/picture reader
- Utilities, games, articles, art
- Our shameless emotional pitch

Order:  
**JUMPDISK SAMPLER**  
 1493 Mt. View Ave.  
 Chico, CA 95926



JUMPDISK has been published every month since August 1986. Without fail. We ship orders day received.

Questions? Call us at (916) 343-7658  
 Dealers, get in touch. JUMPDISK sells!

\*Amiga is a registered trademark of Commodore-Amiga Inc.

Thank you for reading our ad carefully!

## LIONHEART

### BUSINESS & STATISTICAL SOFTWARE

Explanatory books with professional compiled software; the new standard for statistical use. The influential Seybold Report on Professional Computing has this to say about Lionheart "...our sentimental favorite because of its pragmatic approach to the basic statistical concepts... The thinking is that the computer merely facilitates the calculations; the important thing is to be able to formulate a problem correctly and to determine what type of analysis will be most valuable. Let Lionheart help you get ahead of the competition!

- EXPERIMENTAL STATISTICS.....\$145
- BUSINESS STATISTICS..... 145
- MARKETING STATISTICS..... 145
- QUALITY CONTROL..... 145
- BIOMETRICS..... 145
- ECONOMETRICS..... 145
- FORECASTING AND TIME-SERIES.. 145
- SALES AND MARKET FORECASTING 145
- PROJECT PLANNING..... 145
- DECISION ANALYSIS TECHNIQUES. 145
- OPTIMIZATION..... 145
- LINEAR & NON-LINEAR PROGRAMMING 95
- MULTIVARIATE ANALYSIS..... 125
- REGRESSION..... 95

SEND FOR FREE BROCHURE

P.O. BOX 379, ALBURG, VT 05440  
 (514) 933-4918

(continued on page 59)



# Expanding Reference

*Expanding reference* is not just an empty promise. The pages of *Amazing Computing™* are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of *Amazing Computing's* Back Issues contains articles ranging from building your own IBM Disk controller, to setting up your own startup sequence. *Amazing Computing™* has repeatedly been the first magazine to offer the Amiga users solid, in depth reviews and hands on articles for their machines.

## From the Beginning

Since February 1986, *Amazing Computing™* has been providing users with complete information for their Amiga. This storehouse of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga any user will find useful. AC was the first magazine to document CLI, tell its readers how to connect a 5 1/4 IBM drive, describe a 1 meg upgrade hardware project for the A1000, and many more. Please read the list of topics AC has covered below to find the information you have been missing.

## Back Issues are \$5.00 US, \$6.00 Canada and Mexico, \$7.00 Foreign Surface

All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.

## Limited Supply

Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Complete your *Amazing Computing™* library today while these issues are still available. Complete the order form in the back of this issue and mail with your check or money order to:

Back Issues  
Amazing Computing™  
PiM Publications, Inc.  
P.O.Box 869  
Fall River, MA 02722

### Volume 1 Number 1 Premiere 1986

Super Sphere By Kelly Kauffman An ASBasic Graphics prog.  
Date Virus By J. Foust A disease may attack your Amiga!  
EZ-Term By Kelly Kauffman An ASBasic Terminal program  
Mega Menu By P. Kivowitz Programming files & mouse care  
Inside CLI By G. Musser a guided insight into the AmigaDOS™  
CLI Summary By G. Musser Jr. A list of CLI commands  
AmigaForum By B. Lubkin Visit CompuServe's Amiga SIG  
Commodore Amiga Development Program By D. Hicks  
Amiga Products A listing of present and expected products

### Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA  
Inside CLI: part two G. Musser Investigates CLI & ED  
A Summary of ED Commands  
Live! by Rick Miner A review of the Beta version of Live!  
Online and the CTS Fabrite 2424 ADH Modem by J. Foust  
Superman V.1.0 By K. Kauffman A term. prog. in Amiga Basic  
A Workbench "More" Program by Rick Wirth  
Amiga BBS numbers

### Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viverio  
Reviews of Raster, Baratacass and Mindshadow  
Forth! The first of our ongoing tutorial  
Deluxe Draw! by R. Wirth An Amiga Basic art program  
Amiga Basic: A beginners tutorial  
Inside CLI: part 3 by George Musser George gives us PIPE

### Volume 1 Number 4 May 1986

SkyFox and Artiofox Reviewed  
Build your own 5 1/4 Drive Connector By Ernest Viverio  
Amiga Basic Tips by Rick Wirth  
Scripser Part One by P. Kivowitz prog. to print Amiga screen  
Microsoft CD ROM Conference by Jim O'Keefe  
Amiga BBS numbers

### Volume 1 Number 5 1986

The HSI to RGB Conversion Tool  
by S. Pietrowicz Color manipulation in BASIC  
AmigaNotes by Rick Roe The first of the Amiga music columns  
Sidecar A First Look by John Foust A first "under the hood"  
John Foust Talks with R. J. Mical at COMDEX™  
How does Sidecar affect the Transformer  
an interview with Douglas Wyman of Smile  
The Commodore Layouts by J. Foust A look Commodore "cuts"  
Scripser Part Two by P. Kivowitz  
Marsuder reviewed by Rick Wirth  
Building Tools by Daniel Kary

### Volume 1 Number 6 1986

Temple of Apehli Trilogy reviewed by Stephen Pietrowicz  
The Halley Project: A Mission in our Solar System  
reviewed by Stephen Pietrowicz  
Flew: reviewed by Eny Bobo  
Textcraft Plus a First Look by Joe Lowery  
How to start your own Amiga User Group by William Simpson  
Amiga User Groups  
Mailing List by Kelly Kauffman a basic mail list program  
Pointer Image Editor by Stephen Pietrowicz  
Scripser: part three by Perry Kivowitz  
Fun With the Amiga Disk Controller by Thom Sterling  
Optimize Your AmigaBasic Programs for Speed by Pietrowicz

### Volume 1 Number 7 1986

Aegis Draw: CAD comes to the Amiga by Kelly Adams  
Try 3D by Jim Meadows an introduction to 3D graphics  
Aegis Images Animator: a review by Eny Bobo  
Deluxe Video Construction Set reviewed by Joe Lowery  
Window requesters in Amiga Basic by Steve Michel  
ROT by Colin French a 3D graphics editor  
"I C What I Think" Ron Peterson with a few C graphic progs  
Your Menu Strip by B. Casey program Amiga Basic menus  
IFF Bruah to AmigaBasic "BOB" Basic editor by M. Swinger  
Linking C Programs with Assembler Routines  
on the Amiga by Gerald Hull

### Volume 1 Number 8 1986

The University Amiga By G. Gamble  
Amiga's inroads at Washington State University  
MicroEd a look at a one man army for the Amiga  
MicroEd, The Lewis and Clark Expedition reviewed Frizzle  
Scribble Version 2.0 a review  
Computers in the Classroom by Robert Frizzle  
Two for Study by Frizzle Discovery & The Talking Coloring Book  
True Basic reviewed by Brad Grier  
Using your printer with the Amiga  
Marble Madness reviewed by Stephen Pietrowicz  
Using Fonts from AmigaBasic by Tim Jones  
Screen SeVer by P. Kivowitz A monitor protection prog. in C  
Lattice MAKE Utility reviewed by Scott P. Evenden  
A Tale of Three EMACS by Steve Poling  
Jamez File Reader in Amiga Basic by T. Jones

### Volume 1 Number 9 1986

Instant Music Reviewed by Steve Pietrowicz  
Mindwalker Reviewed by Richard Knipper  
The Aegis Memory Board Reviewed by Rick Wirth  
TxEd Reviewed by Jan and Giff Kent  
Amazing Directory A guide to the sources and resources

### Volume 1 Number 9 1986 continued.

Amiga Developers A listing of Suppliers and Developers  
Public Domain Catalog A listing of Amicus and Fred Fish PDS  
Doe 2 Doe review R. Knipper  
Transfer files from PCMS-DOS and AmigaBasic  
MaxiPlan review by Richard Knipper The Amiga Spreadsheet  
Gizmoz by reviewed by Peter Wayner Amiga extras!  
The Loan Information Program by Brian Casey  
basic prog. to for your financial options  
Starting Your Own Amiga Related Business by W. Simpson  
Keep Track of Your Business Usage for Taxes by J. Kummer  
The Abooft Amiga Fortran Compiler reviewed by R. A. Reese  
Using Fonts from AmigaBasic, Part Two by Tim Jones  
8000 Macros on the Amiga by G. Hull Advance your ability.  
TDI Mode-2 Amiga Compiler review by S. Fawcett

### Volume 2 Number 1 1987

What Dig-View is... Or, What Genlock Should Be! by J. Foust  
AmigaBasic Default Colors by Bryan Casey  
AmigaBasic Titles by Bryan Casey  
A Public Domain Module-2 System reviewed by Warren Block  
One Drive Compile by Douglas Lovell  
Using Lattice C with a single drive system  
A Megabyte Without Megabucks by Chris Irving  
An Internal Megabyte upgrade  
Dig-View reviewed by Ed Jakober  
Defender of the Crown reviewed by Keith Conforti  
Leader Board reviewed by Chuck Raudonis  
Roundhill Computer System's PANEL reviewed by Ray Lance  
Dig-Point...by New Tek reviewed by John Foust  
Deluxe Paint II ...from Electronic Arts reviewed by J. Foust

### Volume 2 Number 2 1987

The Modem by Joseph L. Rothman efforts of a BBS Sysop  
MacroModem reviewed by Stephen R. Pietrowicz  
GEMINI or "It takes two to Tango" by Jim Meadows  
Gaming between machines  
BBS-PCI reviewed by Stephen R. Pietrowicz  
The Trouble with Xmodem by Joseph L. Rothman  
The ACO Project...Graphic Teleconferencing on the Amiga  
by S. R. Pietrowicz  
Flight Simulator II...A Cross Country Tutorial by John Refferty  
A Disk Librarian in AmigaBASIC by John Kennan  
Creating and Using Amiga Workbench Icons by C. Hanel  
AmigaDOS version 1.2 by Clifford Kent  
The Amazing MIDI Interface build your own by Richard Rae  
AmigaDOS Operating System Calls and  
Disk File Management by D. Haynie  
Working with the Workbench by Louis A. Mamakos Prog in C

### Volume 2 Number 3

The Amiga 2000™ by J. Foust  
A First look at the new, high end Amiga™  
The Amiga 500™ by John Foust  
A look at the new, low priced Amiga  
An Analysis of the New Amiga PCs by J. Foust  
Speculation on the New Amigas  
Gemini Part II by Jim Meadows  
The concluding article on two-player games  
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith  
The Winter Consumer Electronics Show by John Foust  
AmigaTrix by W. Block Amiga™ shorts  
Intuition Gadgets by Harriet Maybeck Tolly  
A journey through gadget-land, using C  
Shanghai reviewed by Keith M. Conforti  
Chessmaster 2000 & Chessmate reviewed by Edwin V. Apel, Jr.  
Zing! from Meridian Software reviewed by Ed Berovitz  
Forth! by Jon Bryan Get stereo sound into your Forth programs.  
Assembly Language on the Amiga™ by Chris Martin  
Roomers by the Bando Genlocks are finally shipping, & MORE!!!  
AmigaNotes by R. Rae Hum Busters... No street? Y no?...  
The AMICUS Network by J. Foust  
CES, user group issues and Amiga Expo\*

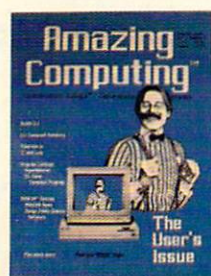
### Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist  
The Mouse That Got Restored by Jerry Hull and Bob Rhode  
Bluething Public Domain Diska with CLI by John Foust  
Highlights: the San Francisco Commodore Show by S. Hull  
Speaker Sessions: San Francisco Commodore Show H Tolly  
Household Inventory System in AmigaBASIC™ by B. Casey  
Secrets of Screen Dumps by Natsun Okun  
Using Function Keys with MicroEmacs by Greg Douglas  
AmigaTrix II by Warren Block More Amiga shorts  
Basic Gadgets by Brian Casey Create gadget functions  
Gridiron reviewed by K. Conforti Real football for the Amiga  
Star Fleet I Version 2.1 reviewed by J. Tracy Amigan Space  
The TIC reviewed by J. Foust Battery powered Clock Calendar  
Metascopes review by H. Tolly An easy-to-use debugger

### Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Bette  
The Future Sound Digitizer by W. Block Applied Vision's SD  
Forth! by J. Bryan comparing JForth and Multi-Forth  
Basic Input by B. Casey AmigaBASIC input routine for use in  
all your programs.  
Writing a SoundScape Module in C by T. Fay Programming with  
MIDI, Amiga and SoundScape by SoundScape author  
Programming in 68000 Assembly Language by C. Martin  
Continuing with Counters & Addressing Modes.

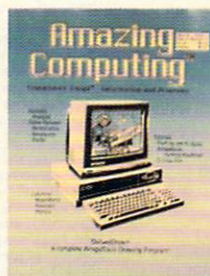




VOLUME 1.1



VOLUME 1.2



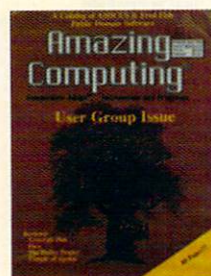
VOLUME 1.3



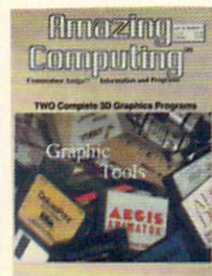
VOLUME 1.4



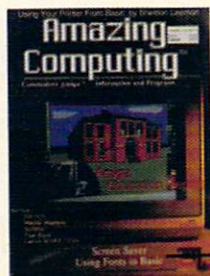
VOLUME 1.5



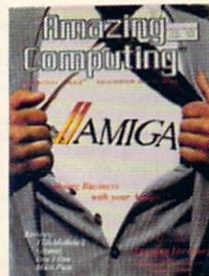
VOLUME 1.6



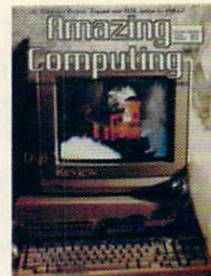
VOLUME 1.7



VOLUME 1.8



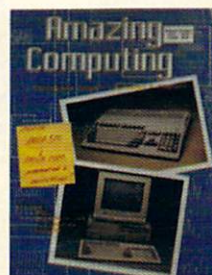
VOLUME 1.9



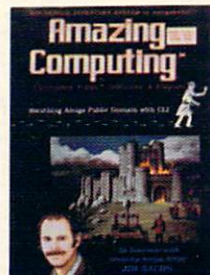
VOLUME 2.1



VOLUME 2.2



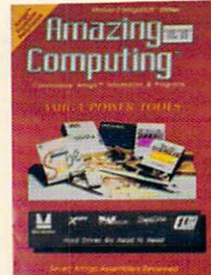
VOLUME 2.3



VOLUME 2.4



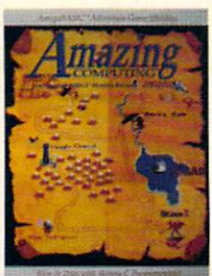
VOLUME 2.5



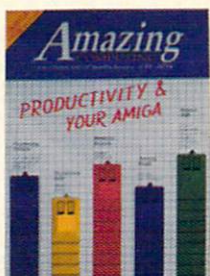
VOLUME 2.6



VOLUME 2.7



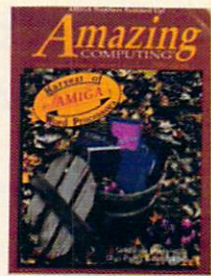
VOLUME 2.8



VOLUME 2.9



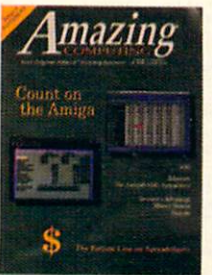
VOLUME 2.10



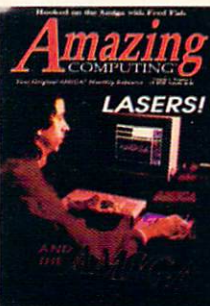
VOLUME 2.11



VOLUME 2.12



VOLUME 3.1



VOLUME 3.2



VOLUME 3.3



VOLUME 3.4



Using FutureSound with AmigaBASIC by J. Meadows  
 AmigaBASIC Programming utility with real, digitized STEREO  
 AmigaNotes Rich Rae reviews SoundScape Sound Sampler.  
 More AmigaNotes by R. Rae A further look at Perfect Sound.  
 Waveform Workshop in AmigaBASIC by J. Shields edit & save  
 waveform for use in other AmigaBASIC programs.  
 The Mimerics Pro MIDI Studio by Sullivan, Jeffrey  
 A review of Mimerics' music editor/player.  
 Intuition Gadgets Part II by H. Maybeck/Tolly Boolean gadgets  
 provide the user with an on/off user interface.

## Volume 2 Number 6 1987

Forth by J. Bryan Access resources in the ROM kernel.  
 The Amazing Computing Hard Disk Review by J. Foust & S.  
 Leemon In-depth looks at the C.Ltd. Hard Drive, Microbotics'  
 MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive and  
 Xebec's 9720H Hard Drive. Also, a look at disk driver software  
 currently under development.

Module-2 AmigaDOS Utilities by S. Fawciszewski A  
 Calls to AmigaDOS and the ROM kernel.  
 Amiga Expansion Peripheral by J. Foust  
 Explanation of Amiga expansion peripherals.  
 Amiga Technical Support by J. Foust  
 How and where to get Amiga tech support.  
 Goodbye Los Gatos by J. Foust Closing Los Gatos.  
 The Amiga Network by J. Foust West Coast Computer Faire.  
 Metacosm Shell and Toolkit by J. Foust A review.  
 The Magic Sac by J. Foust Run Mac programs on your Amiga.  
 What You Should Know Before Choosing an Amiga 1000  
 Expansion Device by S. Grant  
 7 Assemblers for the Amiga by G. Hull Choose your assembler  
 Shakeup Replaces Top Management at Commodore by S. Hull  
 Peter J. Baccarini by S. Hull Manager at IBM gives an inside look  
 Logixis A review by Richard Knepper  
 Organize by A review Richard Knepper database.  
 68000 Assembly Language Programming on the Amiga  
 by Chris Martin  
 Superbase Personal Relational Database by Ray McCabe  
 AmigaNotes by Rae, Richard A look at FutureSound  
 Commodore Shows the Amiga 2000 and 500 at the Boston  
 Computer Society by H. Maybeck Tolly

## Volume 2, Number 7 1987

New Breed of Video Products by John Foust...  
 Very Wild! by Tim Grantman...  
 Video and Your Amiga by Oran Sands III  
 Amigas & Weather Forecasting by Brenden Larson  
 A-Squared and the Live! Video Digitizer by John Foust  
 Aegis Animator Scripts and Cel Animation by John Foust  
 Quality Video from a Quality Computer by Oran Sands III  
 Is IFF Really a Standard? by John Foust...  
 Amazing Stories and the Amiga by John Foust  
 All about Printer Drivers by Richard Beak  
 Intuition Gadgets by Harriet Maybeck Tolly.  
 Deluxe Video 1.2 by Bob Eiler  
 Pro Video C01 by Oran Sands III  
 Dig-View 2.0 Digitizer/Software by Jennifer M. Janik  
 Prism HAM Editor from Implus by Jennifer M. Janik  
 Easy drawing tablet by John Foust...  
 CSA's Turbo-Amiga Tower by Alfred Abuto  
 68000 Assembly Language by Chris Martin.

## Volume 2, Number 8 1987

This month's Amazing Computing focuses on entertainment  
 packages for the Amiga. Amazing game reviews...

SDL Earl Weaver Baseball, Portal, The Surgeon, Little Computer  
 People, Sirbad, StarGlider, King's Quest III and IV, Fiery Tale  
 Adventure, Ultima III, Facets of Adventure, Video Vegas and  
 Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Roomers,  
 Module-2, 68000 Assembly Language and The Amiga Network.  
 Disk-2-Disk by Matthew Leech  
 The ColorFont Standard by John Foust  
 Skinny C Programs by Robert Ramer, Jr.  
 Hidden Messages in Your Amiga by John Foust  
 The Consumer Electronics Show and Comdex by J. Foust

## Volume 2 Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer  
 Impact Business Graphics review by Chuck Raudonis  
 Microfiche Filter review by Harv Laser  
 Pageaster review by Rick Wirth  
 Gizmo Productivity Set 2.0 review by Bob Eiler  
 Kickwork review by Harv Laser  
 Diga Telecommunications Package review by Steve Hull  
 Mouse Time and Timesaver review by John Foust  
 Insider Memory Expansion review by James O'Keane  
 Microbotics Starboard-2 review by S. Fawciszewski  
 Leather Goddesses of Probes  
 reviewed by Harriet Maybeck-Tolly  
 Lattice C Compiler Version 3.10 reviewed by Gary Sarff  
 Manz 3.4a Update reviewed by John Foust  
 AC-BASIC reviewed by Sheldon Leemon  
 AC-BASIC Compiler an alternative comparison by B. Casey  
 Module-2 Programming S. Fawciszewski  
 Raw Console Device Review  
 Directory Listings Under AmigaDOS by Dave Haynie  
 AmigaBASIC Patterns by Brian Casey  
 Programming with Soundscape  
 Todor Fey manipulates samples  
 Bill Volk, Vice-President Angla Development,  
 interviewed by Steve Hull  
 Jim Goodnow, Developer of Manz 'C'  
 interview by Harriet M. Tolly

## Volume 2 Number 10 1987

Max Headroom and the Amiga by John Foust  
 Taking the Perfect Screen Shot by Keith Conforti  
 Amiga Artist: Brian Williams by John Foust  
 Amiga Forum on Computers™ Software Publishing

Conference Transcript by Richard Rae  
 All About Online Conferencing by Richard Rae  
 dBMAN reviewed by Clifford Kent  
 Amiga Pascal reviewed by Michael McNeil  
 AC-BASIC Compiler reviewed by Bryan Casey  
 Bug Bytes by John Steiner  
 Amiga Notes by Richard Rae  
 Roomers by The Bandito  
 68000 Assembly Language by Chris Martin  
 The AMICUS Network by John Foust  
 Amiga Programming:  
 Amiga BASIC Structures by Steve Michel  
 Quick and Dirty Bobs by Michael Swinger  
 Directory Listings Under Amiga-DOS, Part II by Dave Haynie  
 Fast File IO with Module-2 by Steve Fawciszewski  
 Window IO by Read Prodmore

## Volume 2 Number 11 1987

Word Processors Run Down by Geoff Gamble  
 ProWrite, Scribble, and WordPerfect compared  
 LPD Writer Review by Marion Deland  
 Vizal Writer Review by Harv Laser  
 Axiid Review by Warren Block  
 WordPerfect Review by Harv Laser  
 Jaz Jan Interview by Ed Berovitz  
 The author of StarGlider speaks!  
 Do-It-yourself Improvements to the Amiga Genlock  
 Dig-Paint Review by Harv Laser  
 Sculpt 3D Review by Steve Pietrowicz  
 Shadowgate Review by Linda Kasan  
 TeleGames Review by Michael T. Cabral  
 Reason Review: a quick look at an  
 intense grammar examination application  
 As I See It by Eddie Churchill. Peeking at WordPerfect,  
 Gizmo V2.0 and Zngi Keys  
 Bug Bytes by John Steiner  
 AmigaNotes by R. Rae 4 electronic music books  
 Module-2 Programming by Steve Fawciszewski  
 Disks, VOs, and the serial port  
 Roomers by The Bandito  
 68000 Assembly Language by Chris Martin  
 Chris walks through the display routines  
 The AMICUS Network by John Foust  
 Desktop Publishing & Seybold  
 C Animation Part II by Mike Swinger Animation Objects  
 BASIC Text by Brian Casey Pixel perfect text positioning  
 Soundscape Part III by Todor Fey VU Meter and more  
 Fun with Amiga Numbers by Alan Barnett  
 File Browser by Bryan Casey  
 Full Feature BASIC File Browsing utility

## Volume 2 Number 12 1987

The Ultimate Video Accessory by Larry White  
 The Sony Connection by Stewart Cobb  
 15-Puzzles in AmigaBASIC by Zoltan Szepel  
 Life, Part I: The Beginning by Gerald Hull  
 The ultra-complex nine bit solution to the "Game of Life."  
 Amiga Visual by John Foust  
 A new Amiga virus has surfaced. Please check your system.  
 CLI Arguments in C by Paul Castonguay  
 MIDI Interface Adapter by Barry Masoni  
 Amiga 1000-style MIDI interfaces can fit A2000s or 500s  
 Module-2 by Steve Fawciszewski  
 First in a series, a command line calculator in Module-2.  
 AmigaNotes by Richard Rae  
 The audio changes made in the Amiga 500 and 2000.  
 Animation for C Rookies: Part III by M. Swinger  
 tackling double-buffering.  
 The Big Picture by Warren Ring  
 Amiga™ Assembly language programming for the brave  
 Karate Kid Review by Stephen R. Pietrowicz  
 GOI 84 review by John Foust, James O'Keane, and Rick Wirth  
 Three C-64 experts investigate a new Amiga 64 emulator.  
 A-Talk-Plus Review by Brenden Larson  
 "Full-fledged terminal program" & Tektronics capabilities  
 Calligrapher Review by John Foust  
 Animator: Apprentice Review by John Foust  
 Playing Dynamic Drums on the Amiga by David N. Blank  
 WordPerfect Review by Steve Hull  
 Insider/Kickstart Review by Ernest P. Vivalore Sr.  
 RAM & ROM expansion: Comments and installation tips.  
 Bug Bytes by John Steiner  
 Forth by John Bryan  
 DumpPort utility for your Multi-Forth toolbox.  
 As I See It by Eddie Churchill  
 An offset look on Dig-Paint, Portal, and Videoscape 3D.  
 The Amiga Network by John Foust  
 The Commodore Show and AmigaExpo: New York!

## Volume 3 Number 1 1988

AmigaNotes by Richard Rae  
 Digital music generation on the Amiga.  
 C Animation Part IV by Michael Swinger  
 Just when you thought it was safe to go back in the C waters...  
 Forth by John Bryan  
 Sorting out GHP and FAST memory on the Amiga.  
 The Big Picture by Warren Ring During assembler language  
 programming, CLI system calls and manipulating disk files.  
 Bug Bytes by John Steiner  
 Roomers by The Bandito  
 Amiga Dos! 37 6036-based BridgeBoard for the A2000? More!  
 As I See It by Eddie Churchill  
 Opinions, observations, & the birth of a new software generation.  
 68000 Assembly Language Programming by Chris Martin  
 "Create a multi-color screen without using intuition routines!"  
 Module-2 Programming by Steve Fawciszewski  
 A new contender bursts on to the module-2 scene!  
 Amiga Network Special Report: Fall COMDEX by J. Foust  
 Commodore at COMDEX and new products  
 The Ultimate Video Accessory: Part II by Larry White  
 Life: Part II by Gerald Hull  
 "A detailed look at efficient use of the Amiga blitter."  
 FormatMaster: Professional Disk Formatting Engine by G. Martin  
 Put Bash language to work on the drudgery of disk formatting.

BSpread by Brian Casey  
 A full featured AmigaBASIC spreadsheet you can program!  
 AmigaForum Transcript ed. by Richard Rae  
 Zoom in on Commodore's Amiga's Dave Haynie.  
 Halcar Review by Chuck Raudonis  
 "A straightforward, easy to use, functional spreadsheet."  
 VP Professional Review by Suzanne Michel  
 Easy stock portfolio management on the Amiga.  
 Money Mentor Review by Stephen Kemp  
 A personal finance system beyond your checkbook.  
 Investor's Advantage Review by Richard Knepper  
 plus "Poor Man's Guide to the Stock Market."

## Volume 3 Number 2 1988

Laser Light Shows with the Amiga by Patrick Murphy  
 Lasers and the Amiga: A Dazzling Tandem  
 The Ultimate Video Accessory: Part II by Larry White  
 Take the first step toward designing your own videos.  
 Our First Desktop Video by Larry White  
 Step-by-step guide to organizing & presenting your Amiga video.  
 Hooked on the Amiga with Fred Fish by Ed Berovitz  
 Inside views from the man behind all those "Fish" disks.  
 Photo Quality Reproduction with the Amiga and Dig-View  
 by Stephen Lebons  
 Balancing your Checkbook with WordPerfect Macro by S. Hull  
 Hand your checkbook worries over to the Amiga.  
 More Basic Text by Bryan Casey easier text on an Amiga screen  
 Life: Part III by Gerald Hull  
 Sres winds up with famed nine-bit calculation & source to LFER.  
 Solutions to Linear Algebra through Matrix Computations  
 by Robert Ellis  
 Simplify matrix algebra with basic operations & routines.  
 Roomers by Bandito Amiga 3000, Virus news, & Laser Teaser  
 Bug Bytes by John Steiner  
 Module-2 Programming by Steve Fawciszewski  
 Catching up with Celc's source follow-up.  
 68000 Assembly Language Programming by Chris Martin  
 Graphics: Part II of Assembler  
 Arzook's Tomb by Kenneth E. Schaefer  
 "A timely addition to the world of the occult."  
 ART by Steve Fawciszewski  
 An innovative on-base programming language.  
 Forms in Flight by Steve Pietrowicz  
 Render and Animate objects in 3D!  
 Silicon Dreams and the Jewel of Darkness by K. E. Schaefer  
 Laureate suit Larry by Kenneth E. Schaefer  
 Two New Entries From Microbotics by John Foust  
 M501 Expansion & Starboard II Multifunctionboard.  
 Mindlight 7 and People Meter by John Foust  
 Phantasia  
 Ken E. Schaefer with the Amazing Phantasia Character Editor.

## Volume 3 Number 3 1988

Take Five! by Steve Hull  
 Best your frustrations in these five Amiga games.  
 Desktop Video, Part IV by Larry White  
 Put all the pieces together-the desktop video commercial.  
 The Hidden Power of CLI Batch File Processing  
 by Joe Rothman  
 Make your Amiga easier to use with CLI Batch files.  
 A Conference with Eric Graham edited by John Foust  
 The mastermind behind Sculpt 3D and Amiga 3D.  
 Perry Kivowitz Interviewed by Ed Berovitz  
 Amiga insights from a major developer and personality.  
 Jean "Moebius" Giraud Interviewed by Edward L. Fadigan  
 Avant-garde art comes to the Amiga-in a dazzling form.  
 PAL Help by Perry Kivowitz  
 All the help you need for a A1000 expansion reliability.  
 Boolean Function Minimization by Steven M. Hart  
 A useful digital design tool in AmigaBASIC.  
 Amiga Serial Port and Mid Compatibility for Your A2000!  
 by Lynn Ritter and Gary Rentz

Add an A1000-style serial port to your A2000!  
 Electric Network Solutions the Matrix Way by Robert Ellis  
 Engineers' Practice routines for using matrix algebra.  
 In the Public Domain by C.W. Flatts  
 Hot flashes and highlights from the latest Fred Fish disks.  
 The A.M.U.G. BBS List compiled  
 by Joe Rothman, Chet Solace, and Dorothy Dean 100  
 514 BBS phone numbers in the U.S. and Canada.  
 FACC II reviewed by Graham Kinsey  
 Put a freecracker under your floppy drives.  
 Unwilted reviewed by Kenneth E. Schaefer  
 When was the last time a game scared you to death?  
 Flow reviewed by Pamela Rothman  
 Turn your brainstorm into mental works of art.  
 Benchmark Module-2 Compiler reviewed by Richie Beak  
 Program development that beats Pascal to the punch.  
 Bug Bytes by John Steiner  
 Stay abreast of today's bugs and tomorrow's upgrades.  
 Module-2 Programming by Steve Fawciszewski  
 The gameport device and some sprites in action.  
 AmigaNotes by Richard Rae  
 A1000 Create a software-switchable output filter.  
 Roomers by The Bandito  
 Inside AmExpo ... Kickstart 1.4 ... Commodore 'R' Us?  
 The Big Picture by Warren Ring  
 Unwilted arguments to system calls?  
 Discover the Unified Field Theory!

## Volume 3 Number 4 1988

Highlights from AmiExpo, Los Angeles by Steve Hull  
 The Amiga shows off its best in the West.  
 Writing a SoundScape Patch Librarian by Todor Fey  
 Get your hands dirty working within  
 the System Exclusive.  
 Upgrade Your A1000 to A500/2000 Audio Power  
 by Howard Baasen  
 Modifications to help your A1000 make sweet music, too!  
 Amiga Audio Guide  
 Descriptive listing of all Amiga audio products.  
 Guts in Multi-Forth by John Bushakra  
 Push Guts to the limit with these programming tools.  
 Macrobase by Patrick J. Horgan Ease the trauma of

assembly language programming.  
 Amiga Audio Sources The folks behind all those audio products.  
 Take Five! by Steve Hull  
 Four lightning-paced 15es to slash boredom.  
 Amiga Notes by Rick Rae  
 Confounded by sound? Take a basic tour of Amiga audio.  
 The Ultimate Video Accessory, Part V by Larry White  
 Let's add some flash to our video.  
 Bug Bytes by John Steiner  
 The exterminator strikes again.  
 The Big Picture by Warren Ring  
 Part II of the eye-opening Unified Field Theory.  
 Roomers by The Bandito  
 Hardware hijinx ... Toasted video ... the dream Amiga ... and more!  
 In the Public Domain by C.W. Flatts  
 C.W. has hooked the latest Fish disks—here's an inside look.  
 Time Bandit review by Keith Conforti  
 A whole video arcade wrapped up in one game!  
 AudioMaster review by Brandon Larson  
 Friendly digitizing software that samples in real-time.  
 Music Mouse review by J. Henry Lowengard  
 Making music without lifting a finger from the mouse.  
 Amiga-Tax/Canadian Review by Ed Berovitz  
 A Canadian income tax planning, preparation, and  
 analysis package for the Amiga.  
 SAM BASIC review by Bryan Casey  
 A new BASIC which exploits even more unique Amiga features.

## Volume 3 Number 5 1988

Interactive Startup Sequence by Udo Pernitz  
 The Command Line Part I by Rich Falconburg  
 AmigaTrix III by Warren Block  
 Tips and tricks to ease Amiga life  
 Amiga Product Guide: Hardware Edition  
 Proletariat Programming by Patrick Gaud  
 Public domain compiles  
 The Companion by Paul Gosselin  
 The Amiga's Event Handling capability.  
 Mindlight 7 reviewed by David N. Blank  
 Psychodrive 1ad of the 72a updated for the Amiga.  
 VideoScape 3-D 2.0 reviewed by David Hopkins  
 Extend reviewed by Bryan D. Casey  
 An AmigaBASIC extension  
 AssamPro reviewed by Stephen Kemp  
 Opening a door to assembly language programming.  
 APR 68000 reviewed by Roger Nelson  
 Book Reviews by Richard Grace  
 Three 'C' programming texts.  
 CBTREE reviewed by Michael Listman  
 A study collection of functions to aid the C programmer.  
 The Big Picture by Warren Ring  
 The three-part Unified Field Theory winds up  
 Module-2 by Steve Fawciszewski  
 Termination modules for Benchmark and TDI compilers.  
 68000 Assembly Language by Chris Martin  
 Peeling away the complication of display routines.  
 Plus a great collection of monthly columns...

## Volume 3 Number 6 1988

Beer Time reviewed by Steve Carter  
 What makes this inexpensive A1000 battery-backed clock tick?  
 Acquisition reviewed by David N. Blank  
 A look inside the latest release of a powerful relational database.  
 Butcher 2.0 reviewed by Gerald Hull  
 A body collection of diverse image processing utilities.  
 Reassigning Workbench Diska by John Kannan  
 Endless disk swapping comes to a merciful end.  
 Product Guide: Software Tools Edition  
 A listing of all the products you need to put your Amiga to work.  
 An IFF Reader in Multi-Forth by Warren Block  
 Create an easy to use IFF reader in Multi-Forth.  
 Basic Directory Service Program by Bryan Casey  
 A programming alternative to the GimmieZero windows.  
 C Notes from the C Group by Stephen Kemp  
 A beginner's guide to the power of C programming.  
 An Amiga Forum Conference with Jim Mackraz  
 The Amiga market as seen by the "Stepfather of Intuition."  
 Son of Seven Assemblers by Gerald Hull  
 A comparative battle between seven native-code assemblers.  
 The 1988 Commodore Amiga Developers Conference  
 Conferences & new products announced in Washington, D.C.  
 Amiga Working Groups by Perry Kivowitz and Eric Lavitzky  
 An outline of the innovative Amiga Working Groups concept.  
 Take-Five! by Steve Hull  
 An pre-peak at Spring COMDEX and five hot game titles.  
 Hot on the Shelves by Michael T. Cabral  
 New products DOS utilities, image processing, galactic battles,  
 and more!  
 Bug Bytes by John Steiner  
 Bugs check in, but they don't check out.  
 Roomers by The Bandito  
 A3000 hoax, celebs at FAUG, Microsoft comes to the Amiga?  
 AmigaNotes by Rick Rae  
 Amiga audio products move towards the top of the scale.  
 The Command Line by Rich Falconburg  
 Exploring the multi-talented LIST command.

To be continued.....

To Order Back  
 Issues,  
 please use the order  
 form on page 112



## ARexx

### *The REXX Language for the Amiga*

ARexx is a multitasking implementation of the REXX language, an elegant high-level language especially suited for macro-processing. Its clean, simple syntax makes REXX easy to learn for novices — and experienced programmers will appreciate advanced features like compound variables, INTERPRET instruction, and source-level debugging.

- Interactive, Interpreted Operation
- Exceptional String-handling Facilities
- Built-In Library with Over 75 Functions
- Built-In Source-Level Debugger
- Compact, Reentrant code — Only 32K

### **The Emerging Standard**

ARexx is the *de facto* standard for inter-program communication. Software that supports the ARexx command interface can be customized, extended, and combined with other ARexx-compatible software in integrated applications. Watch for announcements of ARexx support from a growing list of Amiga software vendors!

**Available Now ... Only \$49.95**

## WShell

### *The Command Shell You've Always Wanted*

WShell is the best command shell on the market, with features and convenience of use that go well beyond the CLI and 1.3 AmigaShell. And yet it's highly compatible with the CLI, so you don't have to learn a new command language to use it!

- Line Editing/Command History/Aliases
- Built-In Commands/Resident Commands
- Prompt String/Window Title "Variables"
- Concurrent Piping
- Transparent Support for ARexx Macros
- Script-bit Support for "Execute" Scripts
- Compact, Reentrant Code — Only 10K

### **ARexx Command Interface**

With WShell you can use REXX-language macro programs like executables — even as filter programs in a piping system. Use the source-level debugging to single-step through a system of concurrently-running "piped" programs!

**Available Now ... Only \$50**

Ask your Amiga dealer for ARexx and WShell, or order them directly from the author. Please include a check or money order plus \$2/item shipping (\$8 for overseas airmail) and any applicable taxes. Dealer inquiries welcome!

William S. Hawes  
P.O. Box 308  
Maynard, MA 01754  
(508) 568-8695

Amiga is a trademark of Commodore-Amiga, Inc.

## **Pre-School**

**Match-it** \$39.95  
Teaches children basic colors and shapes through games and tutorials.  
*The Other Guys, 55 N. Main, Suite 301D, Logan, UT 84321, (801) 753-7620*

**RE-910 Learning the Alphabet** \$29.95  
Teaches children to recognize upper and lower case letters in the proper order.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

**The Talking Coloring Book** \$29.95  
Talks to the child and teaches color recognition. Drawing option allows children to design their own pictures.  
*J.M.H. Software of Minnesota, Inc., 7200 Hemlock Lane, Maple Grove, MN 55369, (612) 424-5464*

## **Reading-Advanced**

**Bible Files on Disks**  
**Vol. 1** \$34.95  
Genesis - Deuteronomy, 2 disks. Each chapter is saved as a separate file to be used with any ASCII format word processor.  
*John 1:1 Graphics, P.O. Box 316, Bellflower, CA 90706*

**Bible Files on Disks**  
**Vol. 2** \$39.95  
Joshua - Esther, 2 disks. Each chapter is saved as a separate file to be used with any ASCII format word processor.  
*John 1:1 Graphics, P.O. Box 316, Bellflower, CA 90706*

**Bible Files on Disks**  
**Vol. 3** \$19.95  
Job - Songs of Solomon, 1 disk. Each chapter is saved as a separate file to be used with any ASCII format word processor.  
*John 1:1 Graphics, P.O. Box 316, Bellflower, CA 90706*

**Bible Files on Disks**  
**Vol. 4** \$34.95  
Isaiah - Malachi, 2 disks. Each chapter is saved as a separate file to be used with any ASCII format word processor.  
*John 1:1 Graphics, P.O. Box 316, Bellflower, CA 90706*

**Bible Files on Disks**  
**Vol. 5** \$39.95  
Matthew - Revelation, 2 disks. Each chapter is saved as a separate file to be used with any ASCII format word processor.  
*John 1:1 Graphics, P.O. Box 316, Bellflower, CA 90706*

**RL-993 In The Promised Land** \$89.95  
5 disks. High school to adult. Stories of the Chosen People and their conquest and settlement of the Promised Land.  
*MicroEd P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

(continued)

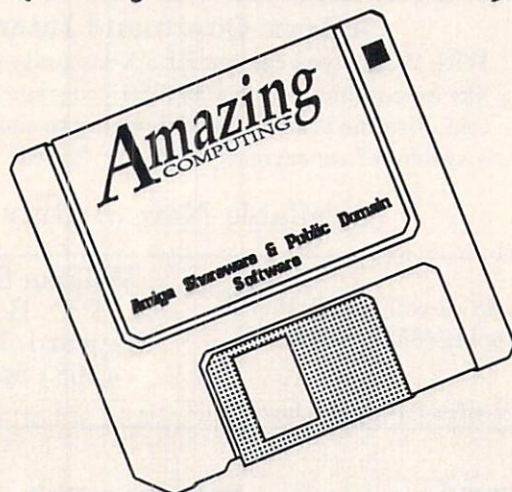


# ANNOUNCING!

## Amazing on Disk

*Source Listings and Executables  
from the pages of Amazing Computing!*

**Only \$6.00 per disk  
(\$7.00 for Non-Subscribers)**



### Now Available: Amazing on Disk #1

Complete source listings from  
Amazing Computing 3.8 and 3.9.

*Featuring:*

**Tumbling Tots** (AC V3.8)

**Modula-2 FFP & IEEE Math Routines** (AC V3.8)

**Gels in Multi-Fort Part II** (AC V3.9)

**The Complete CAI Authoring System  
in AmigaBASIC** (AC V3.9)

*and a few freely redistributable goodies like:*

**Amiga LifeCycles**

The Ultimate Biorythm program for the Amiga.

**MenuEd & VGad** programming tools featured in  
"The Developing Amiga" (AC V3.8.).

**(Order Form on page 112)**

#### VNT

(Visualized New Testament) \$49.95

*Available Soon*

Compatible with ProWrite word  
processor. Red letters, italics, maps.

*John 1:1 Graphics, P.O. Box 316,  
Bellflower, CA 90706*

#### College Aptitude Reading Comprehension

**Exercises** \$65.00

Practice for high-school level readers;  
helps prepare for college aptitude tests.

*Queue, Inc., 562 Boston Ave., Bridgeport,  
CT 06610, (800) 232-2224; (203) 335-  
0908*

#### Complete Lessons in

**Reading and Reasoning** \$149.95

Five disk-set—the complete collection of  
sets I-IV.

*Queue, Inc., 562 Boston Ave., Bridgeport,  
CT 06610, (800) 232-2224; (203) 335-  
0908*

#### Lessons in Reading and Reasoning Sets I-IV

I-III \$39.95; IV \$59.95

Develops critical reasoning skills by  
examining various fallacies.

*Queue, Inc., 562 Boston Ave., Bridgeport,  
CT 06610; (800) 232-2224; (203) 335-  
0908*

#### Reading Adventure 1-3

Adventure 1, \$39.95;

Adventures 2-3, \$59.95

Interactive reading game allows players  
to participate in completing the story.

*Queue, Inc., 562 Boston Ave., Bridgeport,  
CT, 06610, (800) 232-2224; (203) 335-  
0908*

#### Reading and

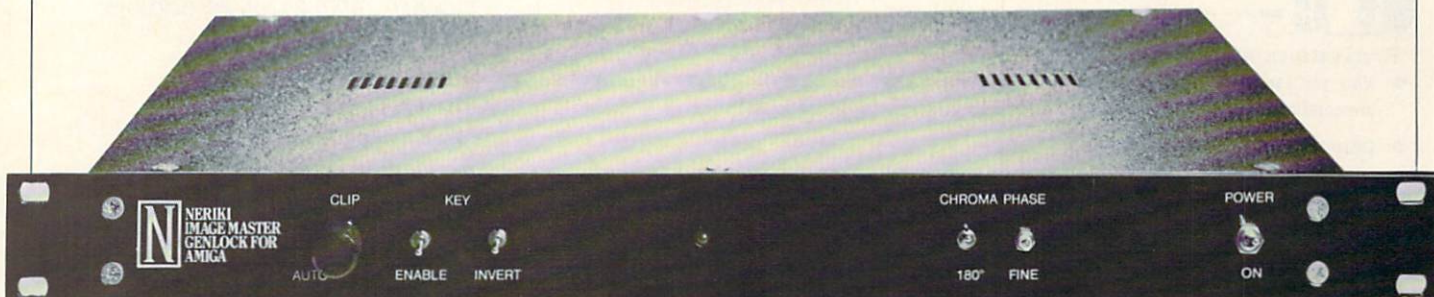
**Thinking I, II, III** \$54.95 each

Reproducible classroom exercises in  
inferential thinking and reading  
comprehension.

*Queue, Inc., 562 Boston Ave., Bridgeport,  
CT 06610, (800) 232-2224; (203) 335-  
0908*



# NERIKI delivers



## a true BROADCAST quality genlock for the AMIGA™

- The **NERIKI IMAGE MASTER™ PRO GENLOCK** outputs 520 lines of encoded NTSC resolution (600 lines in PAL). It incorporates an adjustable luminance keyer and has a chroma phase control.
- The NERIKI supports all graphics modes & resolutions and is compatible with all AMIGA models.
- This rack mountable unit supplies a full complement of BNC input & output connectors, integrating all your professional and broadcast studio requirements.

North American distributor: **COMPU ART**  
P.O. Box 712, Victoria Station,  
Montreal, Qc., Canada H3Z 2V8  
(514) 483-2080 © (514) 737-5865

International Distributor: **TELMAC PTY LTD**  
Unit 12, 126 Queens Road  
Five Dock, N.S.W. 2046,  
Australia 2065 © (02) 745-3466

Neriki Image Master is a trademark of Neriki Computer Graphics PTY Ltd. Amiga is a trademark of Commodore Business Machines Ltd.

### Reading-Elementary

#### Aesop's Fables

\$49.95

12 illustrated stories with reading and vocabulary questions, 800 word spelling list. Ages 6-9, 1-2 players.

*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas NV 89121, (702) 737-8862*

#### Robot Readers

\$29.95 each

Reading game helps children learn to read. Ugly Duckling, Three Little Pigs, Aesop's Fables, Little Red Hen, Chicken Little.

*Hilton Android, P.O. Box 7437, Huntington Beach, CA 92615, (714) 963-4584*

#### RE-915

##### Beginning Reading Skills \$89.95

4 Disks, grades K-3. Over 1000 words in sentences of varying difficulty.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

#### RL-901 In the Beginning \$59.95

Elementary level. The biblical story of the creation of the world. From chapter I of the book of Genesis.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

#### RL-902 Adam and Eve \$29.95

Elementary level. The biblical story of the fall of Adam and Eve. From chapters II and III of Genesis.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

#### RL-904 Abraham, Friend of God

\$29.95

Elementary levels. The story of Abraham's trust and God's promise. From Genesis, chapter XII.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

#### RL-932

##### The First Christmas

\$29.95

Elementary level. The New Testament story of the birth of Jesus in Bethlehem.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

#### RL-938 The Story

##### of the Good Samaritan \$29.95

Elementary level. The parable that showed the importance of treating others as we would want to be treated ourselves.

*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

#### Magical Myths

\$49.95

3 stories with graphics, glossary, and questions. Several games, mini word processor. Ages 8-12, 1-2 players.

*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV 89121, (702) 737-8862*

#### Read & Rhyme

\$49.95

Talking language arts activities in phonics, reading, rhyming and more. Outer-space theme. Ages 5-10, 1-2 players.

*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas, NV 89121, (702) 737-8862*

(continued)



## 35MM SLIDES FROM YOUR ARTWORK!



### Professional 35mm Slides

Now you can have reproduction and presentation quality slides of your work

Distortion-free—fills in raster lines  
crisp bright colors, converts all IFF files

Now Custom graphic art and illustration.

\$10 each for your 1st to 4th slides.

5 to 9 slides—\$8.50

Over 10 slides—\$8.00

Add \$2.00 for shipping.

New York residents add sales tax.

Call (212) 777-7609 FOR DETAILS

Ask for Irene—or write TRU-IMAGE

P.O. Box 660, Cooper Station

New York, N.Y. 10276

Amiga is a trademark of Commodore-Amiga.

### Read-a-Rama \$59.95

Talking program with 6 language arts activities. Includes authoring system for parents to create lessons. Ages 5-8. 2 disks.

Unicorn Software Company, 2950 E.

Flamingo Road, Suite B, Las Vegas NV

89121, (702) 737-8862

### Tales From

#### The Arabian Nights \$49.95

3 stories with graphics, questions, and glossary. Several games, mini word processor. Ages 8-12, 1-2 players. 2 disks.

Unicorn Software Company, 2950 E.

Flamingo Road, Suite B, Las Vegas NV

89121, (702) 737-8862

### The Wonderful

#### Animal Kingdom \$49.95

6 activities teach about the animal kingdom and develop reading and vocabulary skills. 1-4 players, graphics, voice. Age 6-12.

Unicorn Software Company, 2950 E.

Flamingo Road, Suite B, Las Vegas NV

89121, (702) 737-8862

## Science

#### SC-935 Planet Probe \$29.95

Grades 3-9. Teaches facts about the solar system through an arcade-style space simulation.

MicroEd, P.O. Box 24750, Edina, MN

55424, (612) 929-2242

#### Science Expansion Disk

##### for Discovery \$19.95

Grades 9-12. For education or SAT preparations. Covers general science. Requires Discovery Program Disk.

MicroIllusions, 17408 Chatsworth St.,

Granada Hills, CA 91344, (818) 360-

3715; (800) 522-2041

#### The Planetarium \$69.95

See over 9,000 stars to 7th magnitude.

Includes latest NASA ephemerides. Many features.

MicroIllusions, 17408 Chatsworth

St., Granada Hills, CA 91344, (818) 360-

3715; (800) 522-2041

## Spelling/Vocabulary

#### Analogies I-II \$60 each

Analyze, solve, and practice analogies to improve scores on college aptitude tests.

Queue, Inc., 562 Boston Ave., Bridgeport,

CT 06610, (800) 232-2224; (203) 335-

0908

#### Antonyms \$34.95

Prepares students for college aptitude tests by identifying and explaining antonyms through examples and drills.

Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908

#### Discovery Game Disk

##### with Math or Spell \$39.95

An educational space adventure game.

MicroIllusions, 17408 Chatsworth St.,

Granada Hills, CA 91344, (818) 360-

3715; (800) 522-2041

#### How to Spell \$39.95

A tutorial on the rules of spelling.

Queue, Inc., 562 Boston Ave., Bridgeport,

CT 06610, (800) 232-2224; (203) 335-

0908

#### Practical Vocabulary \$54.95

High-school level vocabulary drill program for recognition and use of definitions, antonyms, synonyms, word roots, prefixes.

Queue, Inc., 562 Boston Ave., Bridgeport,

CT 06610, (800) 232-2224; (203) 335-

0908

#### Spelling 1 Expansion Disk

##### for Discovery \$19.95

Spelling game for grades 1 through 10.

Requires Discovery program disk.

MicroIllusions, 17408 Chatsworth St.,

Granada Hills, CA 91344, (818) 360-

3715; (800) 522-2041

#### Spelling 2 Expansion Disk

##### for Discovery \$19.95

Grades 9 through 12. Spelling

proficiency advances through ten levels.

Requires Discovery program Disk.

MicroIllusions, 17408 Chatsworth St.,

Granada Hills, CA 91344, (818) 360-

3715; (800) 522-2041

#### SP-902 to SP-906

##### Spelling \$29.95 each

Grades 2-6. 18 programs per grade-level disk. 360 words, spelling and pronunciation.

MicroEd, P.O. Box 24750, Edina, MN

55424, (612) 929-2242

# APL.68000

# \$99

A HIGHLY OPTIMIZED ASSEMBLER BASED APL INTERPRETER FOR FAST AND POWERFUL PROGRAMS. FEATURES A COMPLETE INTERFACE TO THE AMIGA ENVIRONMENT WITH PULL-DOWN MENUS, REQUESTER AND ALERT BOXES, SPEECH, SOUND AND GRAPHICS FACILITIES.

Order direct for \$99 + \$7 shipping, \$10 Canada. VISA/MC/AMEX + 4% NJ res + 6% sales tax.

## SPENCER

### ORGANIZATION, INC.

P.O. Box 248 Westwood, N.J. 07675

(201) 666-6011



# AMIGA & COMMODORE 64/128 COMPUTERS USERS SHOW

## Boston Bayside Expo Center

200 Mt. Vernon St.,  
Boston, MA

Exit 15 off S.E. Exp. Follow Signs  
Sat., Sept. 10, 88 10-6pm  
Sun., Sept. 11, 88 10-5pm

- Animation & CAD
- Business & Database
- Desktop Publishing
- Hard Drives
- Music Software
- Games & Entertainment
- Spreadsheets
- Hardware, Software & More!!

**One day Adm.**  
Exh. Only \$10.00  
w/ Lectures \$15.00  
**Two Day Adm.**  
Exh. Only \$15.00  
w/ Lectures \$25.00  
For info: 1-800-344-3773  
In CA: (415) 388-8893

**Exhibits • Lectures • Classes • Sales**

**D - Five Associates**

19 Crosby Drive  
Bedford, MA  
01730-0523

(617) 275-8892

Tired of the high cost of computer repairs?

→ **FLAT** Labor charges

→ **FREE** Estimates


→ **Warranty** work

Also:


1764 to 512K: \$61.95

128 64K vdc RAM: \$40.00

NEW: C=1902 conversion to RGB-I: \$40.00




C-64/128 & peripherals




Amiga 1000

\* plus parts and sales tax

Authorized  
Commodore  
Service  
Center





Commodore PC-10

## SP-907

### Spelling Detective Game \$39.95

Learn how to spell the 100 most frequently misspelled words. Also teaches the use of phonemes.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### The Logic Master \$49.95

Helps develop high-level thinking skills in analogies and number series. Includes fantasy role playing game. Grades 5-12.  
*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas NV 89121, (702) 737-8862*

## The Word Master

### Vocabulary Builder \$49.95

4 educational activities designed to build word-power skills, including placement test and Pac-Man type maze game.  
*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas NV 89121, (702) 737-8862*

## VO-920

### Vocabulary Series \$49.95

High level. Words from New York Times, Atlantic Monthly, etc. Learner controlled instruction.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

## Vocabulary

### Adventure I-III \$59.95

Players earn treasures by answering challenging vocabulary questions.  
*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

### WD-915 Word Demons \$29.95

A lesson in usage of homonyms and verbs.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

## Social Studies/Geography

### All About America \$59.95

U.S. history and geography with 16 stories, illustrations, questions, maps and quizzes. Ages 6-11. 2 disks.  
*Unicorn Software Company, 2950 E. Flamingo Road, Suite B, Las Vegas NV 89121, (702) 737-8862*

### Geography Expansion Disk

#### for Discovery \$19.95

Physical and political geography and more. Grades 9 through 12. Requires Discovery program disk.  
*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

### Social Studies Expansion Disk for Discovery \$19.95

Politics, religion, population growth and more. For grades 9 through 12. Requires Discovery program disk.  
*MicroIllusions, 17408 Chatsworth St., Granada Hills, CA 91344, (818) 360-3715; (800) 522-2041*

## SS-909 Social Studies

### Vocabulary \$29.95

Elementary level. Social studies vocabulary in farming, forest, deserts, ocean, manufacturing, trading and political fields.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### SS-956 Introducing Maps \$59.95

2 disks. Digitized pictures. Improve students' knowledge of geography.  
*MicroEd, P.O. Box 24750, Edina, MN 55424, (612) 929-2242*

### Storybook Capitals \$24.95

A 2-disk set.  
*Classic Concepts Futureware, P.O. Box 786, Bellingham, WA 98227, (206) 733-8342*

## U.S. & World Geography

### Adventure \$59.95 each

Game teaches student to identify geographical features of the U.S. & world.  
*Queue, Inc., 562 Boston Ave., Bridgeport, CT 06610, (800) 232-2224; (203) 335-0908*

(see Vendor List on page 64)



# HIGHER PERFORMANCE...AND CHEAPER TO BOOT!

**FData-10** Single 3.5" External Drive ..... **\$149.95**

**FData-20** Dual 3.5" External Drive w/Power Supply ..... **\$299.95**

- Fully 1010 Compatible
- Ultra Compact
- Acoustically Quiet
- Amiga® Color Coordinated
- Daisy Chainable
- Extra Long Cable
- High Performance
- Super Low Price

POLICY: Shipping and handling extra. Personal and company checks require 3 weeks to clear. For faster delivery, use your credit card or send cashier's check or bank money order. Credit cards are not charged until we ship. All prices are U.S.A. prices and are subject to change, and all items are subject to availability. These prices reflect a 5% cash discount. For all credit card purchases there will be an additional 5% charge. Defective software will be replaced with the same item only. All sales are final and returned shipments are subject to a restocking fee.

**FLEXIBLE DATA  
SYSTEMS, INC.**

10503 FOREST LANE • SUITE 148 • DALLAS, TX 75243  
FAX: 214-669-0021

**214-669-3999**

Amiga® is a registered trademark of Commodore-Amiga, Inc.



## Vendor List

### Artworx

1844 Penfield Rd.  
Penfield, NY 14526  
(716) 385-6120; (800) 828-6573

### Associated Computer Services

1306 E. Sunshine  
Springfield, MO 65804  
(417) 887-7373

### Classic Concepts Futureware

P.O. Box 786  
Bellingham, WA 98227  
(206) 733-8342

### Electronic Arts

1820 Gateway Dr.  
San Mateo, CA 94404  
(800) 245-8525; in CA, (800) 562-1112

### Emerald Intelligence

334 S. State Street  
Ann Arbor, MI 48104  
(313) 663-8757

### Finally Technologies

25 Van Ness Avenue  
San Francisco, CA 94102  
(415) 564-5903; FAX (415) 626-4455

### Hilton Android

P.O. Box 7437  
Huntington Beach, CA 92615  
(714) 963-4584

### J.M.H. Software of Minnesota, Inc.

7200 Hemlock Lane  
Maple Grove, MN 55369  
(612) 424-5464

### John 1:1 Graphics

P.O. Box 316  
Bellflower, CA 90706

### Meggido Enterprises

P.O. Box 3020-191  
Riverside, CA 92519  
(714) 683-5666

### MicroEd

P.O. Box 24750  
Edina, MN 55424  
(612) 929-2242

### MicroIllusions

17408 Chatsworth St.  
Granada Hills, CA 91344  
(818) 360-3715; (800) 522-2041

### Progressive Peripherals & Software

464 Kalamath St.  
Denver, CO 80204  
(303) 825-4144; FAX (303) 893-6938

### Queue, Inc.

562 Boston Ave.  
Bridgeport, CT 06610  
(800) 232-2224; (203) 335-0908

### Seven Seas Software

P.O. Box 411  
Port Townsend, WA 98360  
(206) 385-1956

### Software Toolworks

1 Toolworks Plaza,  
13557 Ventura Boulevard  
Sherman Oaks, CA 91423  
(818) 907-6789

### The Other Guys

55 N. Main, Suite 301D  
Logan, UT 84321  
(801) 753-7620

### True BASIC, Inc.

39 S. Main St.  
Hanover, NH 03755  
(800) TR-BASIC; in NH, (603) 643-3882

### Unicorn Software Company

2950 E. Flamingo Road, Suite B  
Las Vegas, NV 89121  
(702) 737-8862



# C Notes From the C Group

by Stephen Kemp, PLINK: SKEMP

Once you have a basic understanding of the object and data types that can be declared in C, you are ready to move on to operators, expressions, and statements. In conjunction with your data, these three things are used to create the functions and logic of C programs.

Actually, the three are very closely related. An operator is a character, or group of characters, that indicates an "operation." A plus sign (+) is an operator which indicates that addition is to occur. Operators are useless without operands. An operand can be a data variable, a constant, a function call, or even an expression. Expressions are composed of one or more operands joined by an operator.  $C = A + B$  is an example of an expression, where A, B and C are variable operands. Statements are made up of one or more expressions followed by a semicolon. Thus, " $C = A + B;$ " is a valid C statement.

As the previous paragraph indicates, operators are the heart of almost every expression and statement. (Note: I said, "almost every expression" because some expressions can have an implied operator rather than an explicit one.) There are many operators in C which indicate that "operations" like arithmetic, equivalence checks, or bit manipulation should be performed. Table 1 is a list of the most commonly used operators defined in C with examples.

Operator precedence determines the order in which expressions are evaluated. Table 1 is organized into precedence levels (highest to lowest, from top to the bottom) with each level separated by a divider line. The order within the levels is unimportant, and will be evaluated on a "first-come first-serve" basis.

On the other hand, operands affected by an operator with higher precedence will be evaluated first. For example, the expression  $A + (B - C)$  will be evaluated differently than the expression  $A + B - C$ , although the results from these two expressions are the same. The difference in evaluation is due to the parenthesis, which have higher precedence than the plus or minus sign. The operands inside the parenthesis will be evaluated first, and that result will then be evaluated with the operand outside the parenthesis. By substituting 10, 6, and 4 for A, B, and C, we can verify that the results of these equations are identical.

Equation 1:  $A + (B - C)$

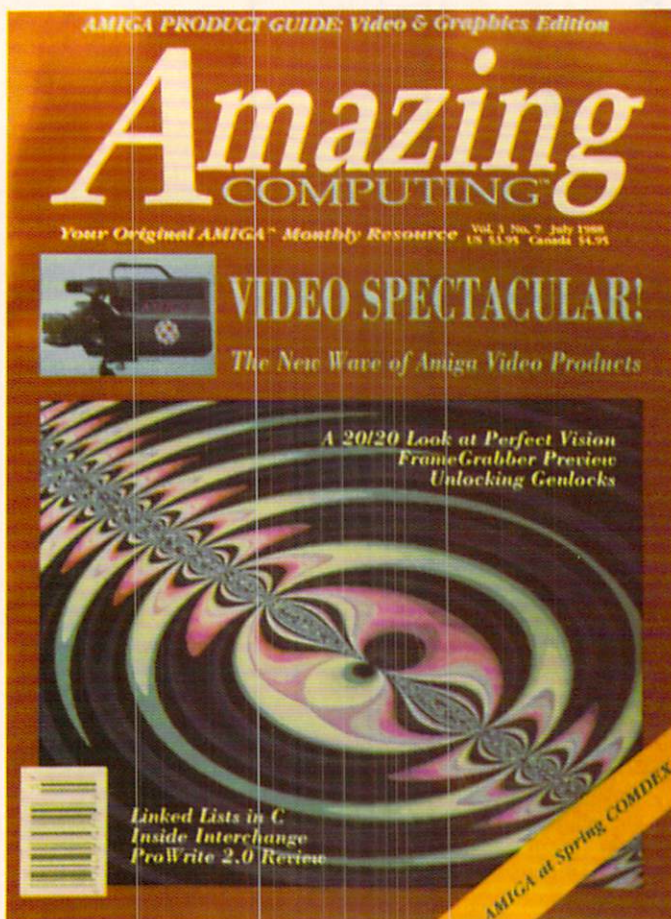
10 + (6 - 4)  
10 + 2  
12

(continued)

Table 1: C Operators by Precedence

Operator	Description	Usage	Associativity
()	Function Call or Group	$(A + B) * C$	Left to Right
[]	Array element	$A[1]$	Left to Right
.	Structure/Union Member	$A.B$	Left to Right
->	Structure/Union Pointer	$A->B$	Left to Right
!	Logical Not	$!A$	Right to Left
~	One's complement	$\sim A$	Right to Left
-	Minus	$A - B$	Right to Left
++	Increment	$A++$	Right to Left
--	Decrement	$A--$	Right to Left
&	Address	$\&A$	Right to Left
*	Indirection (pointer)	$*A$	Right to Left
sizeof	Size in bytes	$\text{sizeof}(A)$	Right to Left
*	Multiplication	$A * B$	Left to Right
/	Division	$A / B$	Left to Right
%	Modulo Division	$A \% B$	Left to Right
+	Addition	$A + B$	Left to Right
-	Subtraction	$A - B$	Left to Right
>>	Shift right	$A >> B$	Left to Right
<<	Shift left	$A << B$	Left to Right
<	Less than	$A < B$	Left to Right
<=	Less than or Equal	$A <= B$	Left to Right
>	Greater than	$A > B$	Left to Right
>=	Greater than or Equal	$A >= B$	Left to Right
==	Equals to	$A == B$	Left to Right
!=	Not Equals to	$A != B$	Left to Right
&	Bitwise AND	$A \& B$	Left to Right
^	Exclusive OR	$A \wedge B$	Left to Right
	Bitwise OR	$A   B$	Left to Right
&&	Logical AND	$A \&\& B$	Left to Right
	Logical OR	$A    B$	Left to Right
?:	Conditional	$(A < B) ? A : B$	Right to Left
=	Set equal to	$A = B$	Right to Left
*=	Multiply with	$A * = B$	Right to Left
/=	Divide with	$A / = B$	Right to Left
%=	Modulo divide with	$A \% = B$	Right to Left
+=	Add with	$A += B$	Right to Left
-=	Subtract with	$A -= B$	Right to Left
<<=	Shift right with	$A >> = B$	Right to Left
>>=	Shift left with	$A << = B$	Right to Left
&=	AND with	$A \& = B$	Right to Left
^=	OR with	$A   = B$	Right to Left
=	Exclusive OR with	$A \wedge = B$	Right to Left
,	Comma	$(A,B)$	Left to Right





## Look What You Missed In AC V3.7 Video Special

### Amazing Features

- Genlock Comparison** by Larry White
- FrameGrabber Preview** by Oran Sands  
Capturing an image can now be as fast as punching a single key!
- A First Look at Interchange** by David Hopkins  
Bridge the gap between those incompatible animation packages.
- Perfect Vision** by Bryan Catley  
Capture, digitize and save pictures from any video source.
- "Anim Man," Gary Bonham** interviewed by Brendan Larson
- Amiga Product Guide: Video/Graphics Edition**  
Thirteen pages devoted to the Amiga's dazzling strong suit.
- Roll Those Presses!** by Barney Schwartz  
Amiga desktop publishing!
- The Amiga at Spring COMDEX in Atlanta**  
by Ed Bercoovitz
- The Developing Amiga** by Steve Pietrowicz  
Developers' notes: PD vs. shareware vs. freely distributable vs. copyrighted.
- Linked Lists in C** by William E. Gammill  
Put dynamic memory to work with linked lists.
- ProWrite 2.0 Review** by Pamela Rothman  
A graphic word processor specializing in efficient editing.
- Doug's Math Aquarium: The Art of Mathematics**  
by Richie Bielak  
Solving equations was never this much fun!
- Bear Products MegaRex II Expansion RAM** by Steve Carter  
An inexpensive way to pump your Amiga up to 2MB.

### Plus our Amazing Columns

- Take Five!** by Steve Hull
- The Command Line** by Rich Falconburg
- Hot on the Shelves** by Michael T. Cabral  
Video music, family trees, IEEE 959 modules, and ... vampires?
- Bug Bytes** by John Steiner
- Amiga Notes** by Rick Rac  
The Other Guys' Synthia: A digital synthesizer.
- C Notes from the C Group** by Stephen Kemp  
Weathering the unknown "C" of basic object and data types.
- Roomers** by The Bandito
- PD Serendipity** by C.W. Flatz

To order Back Issues, please use our order form on page 112.

### Equation 2: A + B - C

```
10 + 6 - 4
16 - 4
12
```

These equations are shown using a type of "long" arithmetic to help you visualize how precedence affects the evaluation process. Notice that in the second equation the + and - operators have the same precedence; thus the operands affected by the first operator are evaluated first. The ( ) in the first equation causes the evaluation to occur differently, yet the outcome of both equations is the same. However, with expressions, the operator precedence would cause the results to be different.

### Equation 3: A - (B + C)

```
10 - (6 + 4)
10 - 10
0
```

### Equation 4: A - B + C

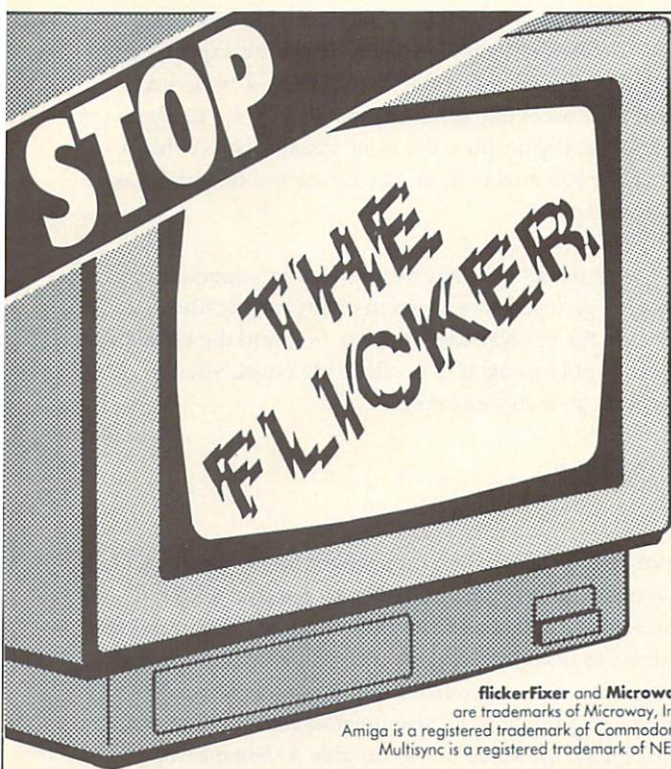
```
10 - 6 + 4
4 + 4
8
```

As you can see in these examples, precedence plays an important role in expression evaluation. Again, as in equation 2, the + and - operators in the second equation have the same

precedence. However, the minus sign now comes first and will be evaluated first. The ( ) in equation 3 causes an evaluation similar to that of equation 1, but unlike the first examples, these equations will not produce the same results. Remember that parenthesis are at the top of the precedence list (see Table 1). When there is any doubt as to how an expression will be evaluated, group the operands with parenthesis.

Operator precedence is important to remember, but so is associativity. Associativity determines the order of evaluation for the operands affected by an individual operator. Left-to-right associativity means that the operand on the left of the operator is evaluated before the operand on the right. Right-to-left associativity is just the opposite. Consider the example,  $Z = A + B * C + D$ . The multiplication has the highest precedence with a left-to-right associativity. The compiler will generate code multiplying B with C first, producing an intermediate result. Next, A will be added to the intermediate result because the + operator also has left-to-right associativity. D will be added last, for the final result of the expression. Since the equals sign has a right-to-left associativity and the lowest precedence, this work must be completed before Z is assigned a value. Using the same numbers and substituting 2 for D, look at this example.





**flickerFixer** eliminates your Amiga 2000's interlace flicker and visible scan lines. The result: **superior quality color or monochrome graphics and text** — for a full range of demanding applications, including CAD, desktop publishing, graphics, animation, and video.

**flickerFixer** fits into the Amiga video slot, is fully compatible with all user software, and does not modify the standard Amiga video signals. The board also upgrades the Amiga 2000 with a flicker free 4096 color palette, has an overscan mode that features a screen size of 704 x 470 pixels and drives most of the popular PC Multisync and VGA monitors, including the NEC Multisync and Mitsubishi XC1429C.

Reviews are impressive: *Amiga GURU 6/88*: "The display is fantastic . . . It is the best display we have ever seen on any computer system." *AMIGAWORLD 5/88*: "Our writers loved it . . . Compare the difference between night and day!"

## *flickerFixer*<sup>TM</sup>

**Advanced Graphics Adapter For The AMIGA® 2000**

**flickerFixer** is priced at \$595. It is made in the USA by Microway, Inc. For more information or to order, call Microway Sales at (508) 746-7341 or your Amiga Dealer.

# Microway

P.O. Box 79  
Kingston, MA 02364 USA  
(508) 746-7341

32 High St.,  
Kings-Upon-Thames, U.K.  
01-541-5466

**NOW FCC  
CLASS B APPROVED**

**Expression:  $Z = A + B * C + D$**

```
Z = 10 + 6 * 4 + 2
Z = 10 + 24 + 2
Z = 34 + 2
Z = 36
36
```

Remember: When in doubt, use parenthesis. Although it would not change the outcome, this last equation would be better written with  $B * C$  inside parenthesis. This would remove any ambiguity that might lead you to misinterpret how your C compiler evaluates the equation.

While you may recognize many of operators in Table 1 from those used in other languages or even from math, there may be several you have not seen before. Discussing each operator would probably be overkill, but I do want to call your attention to some of the operators which have special "properties" in C.

The increment "++" and decrement "--" operators add or subtract 1 from the variable to which they are attached. For example, "var++;" is a valid C statement which would increment the variable "var" by one. Note, however, that we could also have written this statement as "++var;", and achieved the same results. Although this operation seems simple enough to understand, you must use caution when including an expression

using one of these operators inside another expression. The reason is that as an operand in another expression, var++ and ++var are evaluated differently. When one of these operators "follows" a variable name, it is called a "post" operation. That means that the value of the variable will be used in the expression BEFORE the increment/decrement occurs. Likewise, when the operator appears in front of a variable name, it is called a "pre" operation. In this case, the value *after* the increment/decrement is used in the expression. If this seems fuzzy, look at these examples:

**Example 1:**

```
A = 5;
B = A++;
```

**Example 2:**

```
A = 5;
B = ++A;
```

After each of these examples, the value contained in the variable A will be 6, due to the increment operator. In the first example, the value assigned to B will be 5 because it receives the value of A before it is incremented. However, in the second example, the value of B is 6, since it was assigned the value of A after it had been incremented. These examples point out the importance of understanding how to use these "pre" and "post" types of operators.

*(continued)*



Referring back to Table 1, you will notice the ternary operator, "?:". This operator is extremely useful when you want to write C code optimized for size. It does this by giving you the ability to "ask" a true or false statement inside an expression. In fact, it is a simple "if" statement. Although I have not talked about the "if" statement, let's look at an example:

```
if (A > B)
    C = A;
else
    C = B;
```

This is called an if-else statement. (Note: In C, some expressions are evaluated for "true" or "false." True is any non-zero result. False is always equal to 0.) The expression following the word "if" must be enclosed in parentheses. If the expression is determined to be true, then the next statement following the "if" expression is performed. Should the expression prove false, the next statement is skipped. By including the optional "else" statement, we tell the compiler to perform the statement after the else only if the first expression is false. Therefore, we can see that after the if-else statement, C will equal either A or B.

We can accomplish the same results as with the if-else statement by using the ternary operator. The statement would be like this:

```
C = (A > B) ? A : B;
```

Like the "if" statement, this will evaluate whether A is greater than B. If this is a true statement, C will be given the value to the left of the colon, which is A. B is assigned if the expression proves to be false. Now, you may be wondering how this operator can save you code, since the examples I have used would probably generate approximately the same amount of code. However, consider these examples:

#### Example 1.

```
if (A > B)
    my_func(var1,var2,var3,var4,A);
else
    my_func(var1,var2,var3,var4,B);
```

#### Example 2.

```
my_func(var1,var2,var3,var4,(A > B) ? A : B);
```

Function calls in C require more overhead than a simple assignment does. If we used Example 1 in a program, the compiler would generate that overhead twice because of the two function-call statements. The second example, however, would generate the overhead for only one function-call, and the only duplicate overhead would be that required to pass the last variable. Using the second example would make a smaller program while accomplishing exactly the same thing as the first.

Assignment operators are used to assign values to variables. Of course, the most common assignment operator is the equals sign. In C, several additional assignment operators are actually a combination of an equals sign and an arithmetic operation.

These operators were probably included in C because they provide a useful "shorthand" method for writing expressions. Notice the "\*" operator in Table 1. Its usage, A \*= B, is a shorthand method of the same expression, A = A \* B. These two expressions usually produce the same result, unless A has a ++ or -- operator attached to it, in which case the operand A is only evaluated once.

Finally, I want to call your attention to the two operators that will probably give you more grief than any others. These operators are the equivalence operator "==" and the equals sign. When you want to know if two values are equal, you use the "==" operator, as in this example:

```
if (A == B)
    A = 10;
```

The "grief" will happen when on some occasion you forget to type two equal signs—because failing to put two will NOT cause an error. The statement "if (A = B)" is perfectly valid and will evaluate to true or false depending on whether or not B is 0. However, this more than likely will not be what you intended. Not only might the if statement execute improperly, you may also change the value of the variable A. Some compilers will generate a warning on this expression, but most don't because it is not an error. On the other hand, most compilers will produce a warning if you put the equivalence operator where it is unexpected, as in "A == 10;". The difference is that this will cause a true/false evaluation where it isn't required. My best warning this month is to be careful of your equal signs.

To help you understand operators, type in the example program in Listing 1. In past columns, I have made mention of the ASCII character set. This program will print the set on your screen. This program uses many things I have discussed in past columns. Before you begin, I should explain several of the statements used in the program that we have not yet discussed.

```
/* Program ASCII.C
/* This program will print out the ASCII character set by */
/* first defining an array to contain the values and then */
/* printing the array */

main()
{
    char set(128); /* hold the values from 0 - 127 */
    printf("This is the ASCII character set\n\n"); /* a message */

    define_set(set); /* initialize the values in the array */
    print_set(set,10); /* print the ASCII character set */
}

/* end of function main and */
/* end of program */

/* This function will initialize a character array by using */
/* the pointer passed to it as an array */

define_set(p_set)
char *p_set; /* the parameter is a char pointer */
{
    short index; /* define an index variable */

    for(index = 0; index < 128; index++) /* loop through the set */
        p_set[index] = index; /* the index is also the value */
}

/* end of the function define_set */
```



```
/* This function will print the 128 values from indexing */
/* off the pointer passed to it. It will format the output */
/* by printing a carriage return after the number of columns */
/* is reached in each row */
```

```
print_set(p_set,cols)
char *p_set; /* the parameter is a char pointer */
short cols; /* the number of columns per row */
{
    short set_index; /* define an variable for indexing */
    short counter; /* define a column counter variable */

    for(set_index = 0; set_index < 128; ) /* loop without increment */
    {
        for(counter = 1; counter <= cols; counter++, set_index++){
            /* count the cols and increment index */

            if (set_index >= 128) /* if we have reached the end */
                break; /* break out of this loop */

            if (p_set(set_index) < ' ') /* most of the chars before */
                /* the space do not have a */
                /* character representation */
                printf("%d=%c", p_set(set_index), ' ');
            continue; /* continue the loop */
        }

        printf("%d=%c ", p_set(set_index), p_set(set_index));
        /* print the value and char */

    } /* end of inside for loop */

    printf("\n"); /* print a carriage return */
} /* end of outer loop */
} /* end of function print_set */
```

The "for" statements found in two of the functions are looping statements. That means that the statement(s) of the "for" loop

will be performed until: 1) the "for" expression is "satisfied"; 2) or a "break" statement is encountered. The syntax of a "for" statement looks like this:

```
for(expression 1; expression 2; expression 3)
    Statement;
```

Expression 1 is performed only once before the loop begins. Usually, this is where you will initialize the variables used in the loop. Expression 1 is always performed. Expression 2 will be evaluated for true or false. Expression 2 is evaluated before the loop begins the first time. If it is true, the loop will begin or continue. If Expression 2 evaluates false, the "for" loop will end. Expression 3 is performed each time after successfully completing the statement of the for-loop, but before checking Expression 2 again. Expression 3 is usually used to alter the variables used in the loop and/or those used in Expression 2. Any of the components can be omitted, but the semicolons have to appear. If you want more than one Expression 1 or more than one Expression 3, you separate them with commas (not semicolons).

By default, the "for" and "if" statements use the next immediate statement following the expression. We can increase the "scope of control" of these statements using the brace characters {}:

```
if (expression){
    statement;
    statement;
}
```

(continued)

## BEGINNERS

Have you ever seen a GURU MEDITATION ALERT or a TASK HELD REQUESTER telling you that a program has crashed?

If you had GOMF in your system, you would have been able to remove the program that caused the error, and you may not have had to reset your computer!

"GOMF performs flawlessly...[it] is not a luxury, it's a necessity.....perhaps the most useful utility yet introduced for the AMIGA."

-Amiga World

Hypertek/Silicon Springs  
205-2571 Shaughnessy Street  
Port Coquitlam, B.C. CANADA  
V3C 3G3

Dealers: call for special prices!

# GOMF

*the gurubuster !*

NEW VERSION !!!

It's amazing! Now, you can actually save data AFTER a program has crashed! (Even after a total lockup) We call it the GOMF BUTTON.

GOMF BUTTON \$74.95  
GOMF SOFTWARE \$39.95

order toll-free!

1-800-663-8526

Technical Support 604 524-1125

## experts

- \* Restores your memory and system resources after a crash.
- \* Remove tasks, processes and devices
- \* Use from WB or CLI.
- \* Includes pages of technical reference to assist you with your programming.

"Anything that can stop this terrible [GURU] syndrome deserves a round of applause"

- Commodore Magazine

ATTENTION PROGRAMMERS!

Do you have a great idea for a product for the AMIGA? Give us a call for more info!

Other Offerings:

Deluxe MIDI Interface. Simply the best you can buy! Only \$95  
TTL monitor interface \$99



```
for (expression 1; expression 2; expression 3){
    statement;
    statement;
    statement;
}
```

The braces perform much as they do when used in function declarations. The open brace "{" indicates the beginning of influence, and the close brace "}" indicates where the statement ends. Incidentally, the placement of the braces and the indentation of the statement are based on personal coding practice. Good coding practices enhance the maintainability of your code. I could have written the "if" statement like this:

```
if (expression){ statement; statement; }
```

or

```
if (expression)
{
    statement;
    statement;
}
```

I recommend that you indent code that is "under the influence" of other statements. It is also wise to line up the closing brace with the originating statement. The first brace can be placed where you prefer.

Finally, you should notice two other statements used in the program—the "continue" and "break" statements. The continue statement causes the remainder of the statements in the for-loop

to be skipped, and the loop will continue with the next iteration. The break statement will terminate the for-loop whenever it is encountered. Note: If you omit the Expression 2 on a for-loop, a break statement is the only thing that will cause the loop to end (short of terminating the entire program).

As a side note, you will notice that the program excludes trying to print the ASCII values less than the space character. These values are skipped because they usually have no displayable character value. Rather, they are control characters—controlling the display.

The Manx syntax for compiling and linking the program would be:

```
CC ascii.c
LN ascii.o -lc
```

When you run the program, it will print the value and character for each "displayable" member of the ASCII character set on your monitor.

For homework, see if you can figure out how to use the ternary operator "?" to remove one of the printf statements in the last function. (The continue statement will no longer be necessary.) Remember, you can learn a lot from experimenting, so don't be afraid to try something new.

•AC•

## HARD DRIVES FOR AMIGA External Floppy

*Complete Hard Drive Units for the A2000 and A1000/A500*

*A1000 /A500 units come complete with SCSI Host Contoller, case w/power supply and Hard Disk*

*A2000 units come complete with DMA SCSI Contoller , cable and Hard Disk*

### A2000

20 Meg \$539.00

40 Meg \$749.00

65 Meg \$799.00

### A1000 / 500

20 Meg \$639.00

40 Meg \$799.00

65 Meg \$879.00

*3.5" with Metal case, extra length cable, full pass-thru, low power consumption. Fully Compatible with all Amiga Computers.*

Single \$159.95

Dual \$329.95

### *Internal Floppy*

*A2000 only \$139.95*

# COMPUTER MART

YOUR TEXAS SOURCE FOR AMIGA SOFTWARE AND HARDWARE  
IMMEDIATE ACCESS TO OVER 500 PRODUCTS

CALL TOLL FREE

800-443-8236

INFORMATION

409-560-2826

COMPUTER MART 105 LYNN ST. NACOGDOCHES, TEXAS 75961



# COMPUTER AIDED INSTRUCTION (CAI)

## A Generalized Authoring System in AmigaBASIC— Part I

by Paul Castonguay

Computers are making more and more headway in the area of education. A recent example is "The Perfect Score" by Mindscape, a program to help prepare you for the SAT (Scholastic Aptitude Test, required for entrance into most universities). I think the program is excellent, and I recommend it to anyone who needs to learn such material.

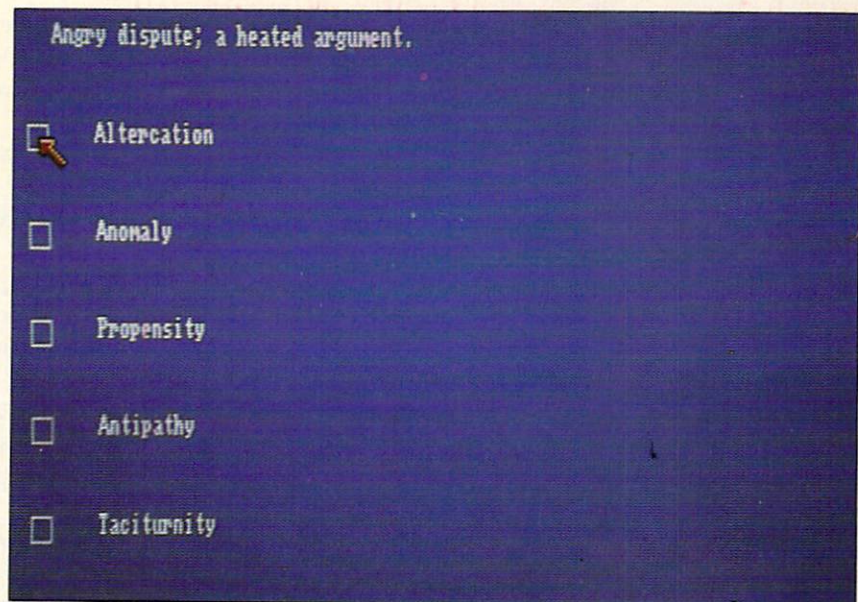
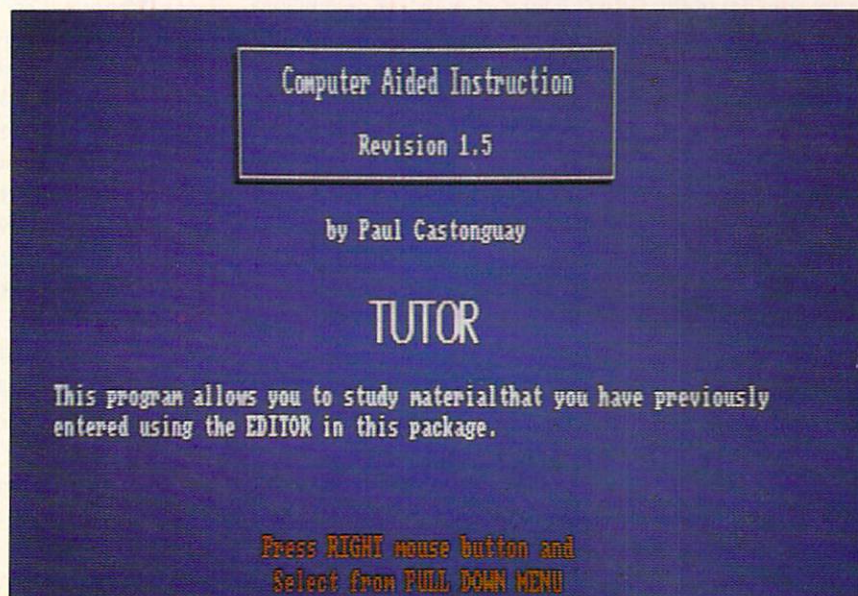
However, it is only good for learning the material contained in its libraries. Wouldn't it be nice to have a program that could help you learn any material you choose? This article presents a program package which allows you to do just that. It consists of an Editor which lets you enter your material, and a program called "Tutor" which lets you play it back in multiple choice format. The result is an efficient computer-aided learning session customized to your needs. I wrote this program to improve my English vocabulary while I was in college, and I found it an efficient studying tool. The Editor allowed me to regularly change material by adding new words I was given in class, and deleting the old words as I learned them. The result was a program which allowed me to sharpen up on words I needed to learn, without wasting time looking at words I already knew.

I wrote these programs in AmigaBASIC so- in the true spirit of computing as a hobby- they could be shared by all users. Everyone gets AmigaBASIC with their machine, so there is nothing to buy. Also, because AmigaBASIC is an interpreted language, the source code is open. You can modify the programs to suit your needs. These programs are a good example of random access data base programming— you may enjoy yourself simply by learning how they work.

### CAI Programs:

Before I go into the details of my programs, some general words on CAI are in order. Can computers really be used to teach people things? Yes, they can! Lots of software today comes with on-line tutorials which teach you, right on your computer, how to use what you have bought. I

(continued)



Vocabulary lessons are just one of the many applications of this authoring system.



have Microsoft Word for my MS-DOS machine and I found its "Learn Program" more educational than struggling through the enormous documentation. Sometimes, however, CAI programs are not so good. Let's talk a bit about that.

The first CAI programs I saw were not very impressive. They were essentially sequential text programs that presented material, in a fixed order, as questions with multiple choice responses. After several sessions with such programs, I realized three problems which seemed to interfere with the learning process. First, I soon learned the correct responses to the first few questions by sequence. The answer to the first question might be multiple response 3, the second might be choice 2, and so on. I could provide many correct responses without even reading the questions. Second, I found myself responding to the position of the correct response in the list of multiple choice responses. I might remember the answer to a particular question was second on the list; I could answer correctly without even reading the answer. Third, many times I could answer a question correctly by eliminating the multiple choice responses that were obviously wrong. There is nothing wrong with that in principle, but in these programs the multiple choice responses never changed. Every time I saw the question, I would answer it by elimination without ever having to read the correct answer.

In this program package, I attempted to resolve these problems by using randomization. There are three levels of randomization. First, all questions are chosen in a random order from the field of questions in a lesson. Second, the position of the correct answer in the list responses is randomized. Third, the multiple choice responses are chosen at random from the entire field of possible responses. Sometimes a question will be easy to answer by elimination because the wrong responses will seem obvious to you. However, the next time you see that question, the responses will be different and you may find it more difficult to eliminate wrong answers. There is more incentive for you to learn the real answer. The program mimics real life in that it challenges you with the same problems in different circumstances. A question that seems easy one time may be more difficult the next time you see it. Playing the program several times is a very rewarding learning experience.

Surely, it takes more than just randomization to create good CAI programs. It also takes good material, and that's where you come in. This program package is a tool. It provides the environment in which computer learning sessions can be created, but it is you who must enter the material. Enter well-organized material, and you will get good results. Enter silly material and you will get silly results. The Editor will prompt you to enter questions and their corresponding answers one at a time. Each question or answer can be as long as three lines of text. A primitive automatic word wrap ensures that no words are split between two lines. Complete lessons are built up by entering as many questions as you like, up to about 1000 per disk. You may have as many lessons as that disk will hold.

Later, when you play back the material using the Tutor program, you will see each question with five multiple choice responses. One of the responses will, of course, be the correct answer you entered for that question when you were in the Editor. The other responses will be answers to other questions in the lesson. You are expected to select the correct answer within the list of five multiple choice responses. If you answer a question correctly, you are congratulated and that question is put aside. If you answer incorrectly, the computer will ask the same question again later during the same session. Thus you will be asked the questions you consistently get wrong! Hopefully, after a few sessions you will eventually learn the material.

There are two general rules you should keep in mind when entering material into the Editor. First, you should not enter in the same lesson two questions which have the same answer. For example, you might enter material from your American History course such that each question is a quotation of an American president and each answer is the corresponding president's name. You could enter in the question window of the Editor something like: "Ask not what your country can do for you, ask what you can do for your country," and in the answer window: "John F. Kennedy." Now suppose you entered as a second question into the same lesson: "By 1980 we will have succeeded in placing a man on the moon," with the corresponding answer "John F. Kennedy." Ahah! Now you will have a problem when you play back the lesson using Tutor. When Tutor asks you either of the above questions, the computer could by chance display two John F. Kennedys in the list of multiple choice responses. You might know the answer to be "John F. Kennedy," but the computer would recognize only one of the John F. Kennedys in the list to be correct. You couldn't know which one to select and even if you knew the correct answer, you would still have a 50% chance of getting it wrong! This doesn't produce a very effective computer-aided learning session. Now the second rule. All questions in a particular lesson should be on the same subject. Entering many questions from unrelated subjects produces lessons in which the questions are too easily answered by elimination. Suppose you entered the material which produced the following question:

*What is the chemical formula for Manganese Dioxide?*

- 1- Abraham Lincoln
- 2- MnO2
- 3- A talkative person.
- 4- Silicon doped with arsenic.
- 5- Veni, vidi, vici.

Responses 1, 3, 4, and 5 are obviously wrong because they are answers to questions from entirely different fields of study. Such a poorly designed lesson would be useless.

Here's one last important point before I get to the programs and how to install them. The lessons produced by this program package are sometimes called "Flash Card Programs," named



after a study method used long before the days of computers. The method consisted of writing questions and answers on opposite sides of standard 5 x 7 index cards. The idea was to read a question from one side of a card chosen at random and then try to answer it without looking at the other side of the card. My program package is nothing more than a high-tech version of the same thing. Note that flash cards (and my program) help you memorize, or sharpen up on material, but they do not explain it. Flash cards are no substitute for a teacher or textbook. They do not expand on the theory behind what you are studying.

For example, in an English vocabulary lesson you might see the words pedagogy and pedestrian and think (with justification) the words are closely related. You might think them synonymous. Wrong! The ped- in pedagogy comes from the Greek root *paidos* meaning child. Combined with -agogy from the root *agogos*, to lead, it means etymologically "the leading of children." The dictionary definition of pedagogy is "the profession of teaching." The ped- in pedestrian, however, comes from the Latin root *pedis* meaning foot. The dictionary definition of pedestrian is "a person on foot or walking." Flash cards cannot teach you such things. There isn't enough room on the card to write all that explanation. However, once you learn such things from a teacher or textbook, flash cards (and my program) are excellent at keeping the knowledge sharp in your mind (examination-ready!).

#### Typing in the Programs

The programs are rather long, but worth typing in. I know everyone has their own way of working with BASIC, but if you're new to this, maybe you'd like to hear how I do it. I prefer to install AmigaBASIC to a blank disk and then use that disk to run AmigaBASIC and save my work. That way, I never make the mistake of saving my work to either the Extras disk, or worse still, the Workbench disk. So here is exactly what I do:

1- I initialize (format) a new blank disk using INTUITION (the Workbench). Single click the disk icon you want to initialize, then choose "Initialize" from the pull-down menu. I usually rename the disk "MyWork.XXX" where XXX is a number of my choice which will allow me to figure out a year from now what is on this disk.

2- I copy (install) AmigaBASIC from a copy of the Extras disk (which came with my computer) by dragging the AmigaBASIC icon from the Extras window to the MyWork window.

3- To enter AmigaBASIC I double-click the AmigaBASIC icon in the MyWork window. Any programs which I save will be on the MyWork disk. If the program is long, it may take several typing sessions to get the job done. Each time I take a break, I save my work. When I return after shutting my computer off, I boot up as usual with a Workbench disk. Then, I insert MyWork in any drive, double-click the MyWork disk icon, double-click the

(continued)

## Another **NEW** Product To Make The **FASTEST** Development Environment Even **FASTER!**

### The **F-BASIC™** Source Level Debugger (SLDB)

#### The Only Debugger Quick Enough For The **F-BASIC™** System

Finally, a full featured Debugger that doesn't require a working knowledge of assembly code. Use the SLDB to quickly debug your F-Basic programs without all those PRINT statements. Check these features of SLDB:

- Debug F-Basic programs at the SOURCE level.
- Fully windowed Intuition interface—windows for Source Code, 68000 Registers, Memory Dumps, Program Variables, etc.
- Set break points and single step trace by either source line or by Assembly Code Line.
- Display all variables, arrays, or RECORDS by name.
- Full reverse-assembler included.
- A comprehensive User's Manual with full documentation of all SLDB supported features and windows.

### The **F-BASIC™** Language System

#### An Enhanced Compiled BASIC Language You've **READ** About It—Now **EXPERIENCE** It!

- ☐ Beginning AMIGA™ programmers choose F-Basic due to its simplicity and ease of use. If you know BASIC, you can use F-Basic immediately.

- ☐ Programmers of all levels choose F-Basic because its hundreds of built-in features provide an AMIGA language environment that coincides with your experience. F-Basic can't be outgrown, due to features like:

- Extensive Control Structures
- LOCAL & GLOBAL Variables
- Ultra Fast Floating Point
- RECORD Structures & Pointers
- INCLUDE & APPEND Separate Files
- PATTERN Matching Support
- Easy high level access to AMIGA Screens, Windows, Menus, Sound, Speech, Graphics and Events.
- Recursive SUBROUTINES & FUNCTIONS
- 32, 16, and 8 Bit INTEGERS
- Powerful String Facilities
- Direct ROM Kernel Access
- Bitwise Operations
- Access to 68000 Registers

- ☐ The F-Basic™ Language System includes:

- Comprehensive User's Manual—Full Documentation of All Supported Features Of The F-Basic System
- Sample Programs Disk—"Hands On" Tutorial
- FastCom™ single pass compiler, which quickly compiles F-Basic programs into assembly code. FastCom is written in F-Basic... That's your guarantee of the speed and efficiency of F-Basic.

The **F-BASIC™** Language System—Only \$79.95  
**F-BASIC™** System and Complete SLDB—Only \$129.95

SEND CHECK OR MONEY ORDER TO:

**DELPHI NOETIC SYSTEMS, INC.**

Post Office Box 7722  
Rapid City, South Dakota 57709-7722  
Credit Card or C.O.D. Call (605) 348-0791

F-BASIC and FastCom are registered trademarks of DNS, Inc.  
AMIGA is a registered trademark of Commodore/AMIGA Inc.



AmigaBASIC icon in the MyWork window, and finally type into AmigaBASIC's output window: LOAD "PROGRAM.NAME". Then I can resume my work where I left off.

### Installing the Programs

When you finally have the two programs typed in completely (or down-loaded from your local BBS), it's time to install some support files you will need to run the programs. This is not difficult, and I will describe every step.

Two files are required because my programs make CALLs to the Amiga's diskfont and graphics libraries. The workbench has some attractive fonts which improve the presentation of programs that you may write. Why not use them? You may also have noticed in the listings of my programs that I used the following statement:

```
CALL Text&(WINDOW(8),SADD(variable$),LEN(variable$))
```

instead of: PRINT variable\$

I do this for speed. My programs display a lot of text on the screen, and if I used BASIC's PRINT statement it would take too long to fill the screen with text. I modified the PRINT statements to CALL Text&O statements and realized a big improvement.

There are also two other files needed by my programs for a simpler reason. You must create two directories (Drawers in INTUITION) to save the questions and answers you enter using the Editor.

OK, now let's get on with it. The special files needed are "graphics.bmap" and "diskfont.bmap"; they are installed using the Extra's disk that came with your computer. Here is what you must do:

1. Insert a working copy of your Extras disk into any drive (df0: for single drive systems).
2. Double-click the Extras icon. The Extras window opens up.
3. Double-click the drawer called BasicDemos. The BasicDemos window opens.
4. Double-click the icon ConvertFD. AmigaBASIC fires up and runs the program. Neat, huh?
5. At the prompt: Enter name of .fd file to read >

Type: Extras:fd1.2/graphics\_lib.fd and press RETURN.

The file you have just entered is the function definition file for all the graphics functions available to AmigaBASIC through the CALL command. The ConvertFD program will convert the graphics\_lib.fd file to a graphics.bmap file which is in the form that AmigaBASIC needs.

6. Now, be sure to type lower case letters where I show them.

At the prompt: Enter name of .bmap file to produce >

On single drive systems type: RAM:graphics.bmap and press RETURN.

On two drive systems with Extras in one drive and MyWork in the other type:

MyWork:graphics.bmap and press RETURN.

There seems to be no response, but be patient. The program takes a minute to give its first screen response and another twenty seconds to finish. You will get three warnings on the screen which do not concern you. The warnings are for programmers and they identify three graphic library functions (LockLayerRom, UnlockLayerRom and AttemptLockLayerRom) which are not available from AmigaBASIC's CALL command. For your own interest, my programs use only the following CALLs: OpenFontO, OpenDiskFontO, SetFontO, MoveO, and TextO. An understanding of these CALLs is totally unnecessary to use my programs. However, you should realize that I use them, because your Amiga may crash if you make a typo in a related program line. There is nothing bad about that; it is simply a clue telling you where to look in the program listing for a typo you may have made. (We all make typos!) When the ConvertFD program finishes, you will see the AmigaBASIC OK prompt in AmigaBASIC's output window. If you have a single drive system, the output of the program was sent to the RAM disk, and if you have a two drive system, it is sent directly to the root directory of your MyWork disk

7. Restart the ConvertFD program by typing RUN in AmigaBASIC's output window and pressing return.

8. At the prompt: Enter name of .fd file to read >

Type: Extras:fd1.2/diskfont\_lib.fd and press RETURN.

9. At the prompt: Enter name of .bmap file to produce >

On single drive systems type: RAM:diskfont.bmap

On two drive systems type: MyWork:diskfont.bmap

You will notice that this time the program finishes in a lightning-fast 5 seconds! You may now leave AmigaBASIC by choosing Quit on MENU or by typing SYSTEM in AmigaBASIC's output window and pressing RETURN. If you have a two drive system skip step 10.

10. If you have a single drive system, insert the disk MyWork and double-click its icon. Also, double-click the RAM icon.



Drag the two .bmap files from the RAM window to the MyWork window. (You may have to drag the RAM window aside to expose the MyWork window underneath).

11. Now double-click the Workbench icon.
12. Drag the drawer called "Empty" from the Workbench window to the MyWork window. Using the Rename item available from the pull-down MENU, rename the Empty drawer on the MyWork disk as "This Disk" (Yes, there's a space between the words "This" and "Disk").
13. Drag a second Empty drawer from the Workbench window to the MyWork window.
14. Rename this second Empty drawer "Lessons".
15. Close all windows on the Intuition's worksurface except the MyWork window.
16. Expand the MyWork window until it is full size.
17. Drag the Editor and Tutor icons to the top left-hand corner of the window.
18. Drag all other icons to the bottom of the window.
19. While holding down the SHIFT key, select all icons in the window. As you select each icon it turns and stays black.
20. While still holding down the SHIFT key, resize the window until only the Editor and Tutor icons are visible.
21. While still holding down the SHIFT key, select the MyWork disk icon.
22. While still holding down the SHIFT key, drag the MyWork window to the center of the screen where you would like it to appear when you use the software.
23. Press the RIGHT mouse button and choose Snapshot from the Special MENU.
24. Release the SHIFT key.
25. Rename this disk CAI.
26. Make a backup of this disk.

If you typed in these programs yourself, you will have to debug them. In the listings of both programs you will notice the statement SYSTEM is used. In the Editor it appears in the "Quit.Editor:" and "Fast.Quit" routines. In the Tutor it appears in the "Terminate:" routine. As the note in the listing suggests, you should change these statements to END. The statement SYSTEM

will cause each program to stop by exiting AmigaBASIC and returning to the INTUITION worksurface when "GOTO Workbench" is selected from the pull-down MENU. The statement END will cause the program to stop, but your Amiga will remain in AmigaBASIC. That will make it easier for you to debug and correct your programs. Later, when you are sure that they are working correctly, you can change these statements back to SYSTEM.

To help those who have typed in the programs by hand, I will now describe most of the operational characteristics in each program. Hopefully, this will help you determine when your programs are error-free.

### **The Editor**

Insert your CAI disk into drive df0 (for single drive systems) or, df1 (for two drive systems). Double-click the CAI disk icon. The CAI window opens, exposing the Editor and Tutor icons. Double-click the Editor icon. If you have a single drive system, you will be asked to swap between the CAI disk and the Workbench disk twice. It is necessary for the program to pick up the graphics and diskfont libraries which reside on the Workbench disk. Obviously, it is more convenient if you have two drives so, you won't have to perform these disk swaps. However, the swaps are worth it. I compliment the Amiga's designers for including a disk operating system that is intelligent enough to remind the user to perform the necessary disk swaps without any extra effort on the programmer's part.

After a "... please wait ..." message is displayed for about 15 seconds, the title page of the editor will appear. Notice the use of Amiga's fancy fonts. If my program cannot find these fonts, you will see a message telling you which font could not be found and the program will halt so you can debug it. Look in the "Initialize:" routine for possible typing errors.

Press and hold the RIGHT mouse button and verify that the items of the pull down menu called "Stuff you can do" are:

Open/Create Lesson	activated	(solid text)
Delete Lesson File	activated	
Quit Lesson	<Q>	de-activated (ghosted text)
Read Question	<R>	de-activated
Add Question	<A>	de-activated
Delete Question	<D>	de-activated
I Changed My Mind		de-activated
GOTO Tutor		activated
GOTO Workbench		activated

First verify that the "GOTO Workbench" item works. If you have changed the SYSTEM statements to END statements as I suggested, the program will stop and you will be in AmigaBASIC. Shrink and drag BASIC's OUTPUT and LIST windows to expose the title bar of the Workbench screen. Point to the title

(continued)



bar, press and hold the LEFT mouse button, and drag the Workbench screen down. Verify there are no windows or screen hidden behind the Workbench screen that may have been left over from the program after it quit. Also, verify that BASIC's four MENU's (Project, Edit, Run, and Windows) are properly restored. Fire up the program again by typing RUN in BASIC's OUTPUT window.

### **The Tutor Program**

You are now in the title page of the Tutor program. Verify that the pull-down MENU contains the following items:

Load Lesson	<L>	activated
Quit Lesson	<Q>	de-activated
GOTO Editor		activated
GOTO Workbench		activated

This time choose "Open/Create Lesson" from the Editor's pull down MENU. You will see the message "Directory of: [This Disk]," a file called ".info," a message "Enter existing or new filename," and finally a blinking cursor. Press the RIGHT mouse button and verify that the only items activated are "GOTO Tutor" and "GOTO Workbench." Now, let's enter some material which will make it easy for you to verify that both the Editor and Tutor are working properly. At the blinking cursor enter the filename "Test." The screen will flash, a beep will sound, and a small window titled "Program Request" will open in the top left corner of the screen displaying the message: "There is no Test, Do you want to CREATE it?" You will also see two selection boxes marked "Yes" and "No." Select the Yes box and you will see the message "...looking for Test ..." followed by the CAI Editor screen. From this screen you may add or delete question/answer pairs in the lesson Test. You should see the name of the lesson and the number of questions previously entered into it at the top of the screen. Verify that the only pull down MENU items activated are: "Add Question," "Delete Question," "Read Question," and "Quit Lesson." Choose Add Question and you will be asked "What question you want to enter" followed by a blinking cursor. Enter 1 and you will see a window, a blinking cursor, and a prompt asking you to enter question 1. Enter: "This is question one." and press RETURN. You will see a second window, a blinking cursor, and a prompt asking you to enter answer 1. Enter: "This is answer one." and press RETURN. Question 1 will be saved to disk and you will be returned to the CAI Editor screen. You have just entered your first question/answer pair in the lesson called Test. Enter the remaining material:

This is question two.	This is answer two.
This is question three.	This is answer three.
This is question four.	This is answer four.
This is question five.	This is answer five.
This is question six.	This is answer six.
This is question seven.	This is answer seven.
This is question eight.	This is answer eight.
This is question nine.	This is answer nine.
This is question ten.	This is answer ten.

Choose "Quit Lesson" from the pull-down MENU and you will be returned to the title page of the Editor. We are not finished testing the Editor, but first we will go to the Tutor program. We will return later to the Editor to see if we can delete questions and lessons files as the program should allow. But first, choose "GOTO Tutor" from the pull-down MENU.

Just as we did when we tested the Editor, choose "GOTO Workbench" from the pull-down MENU and verify that there are no extra windows or screen concealed behind the Workbench screen after the program Tutor closes. Fire up Tutor again and pick "Load Lesson" from the pull-down MENU. You will see "Directory of [This Disk]" and the lesson "Test" listed. That's the lesson you have just entered using the Editor. Enter "Test" at the blinking cursor and you will see one of the questions at the top of the screen, as well as a list of five multiple choice responses. You probably now appreciate why I asked you to enter such simple material. Verify that the only activated item on the pull down MENU is "Quit Lesson." Start answering questions by selecting the answer boxes. Verify that you get the response you would expect. You should play this lesson several times, keeping the following things in mind:

When you answer a question correctly, you will be told how many questions remain in the lesson. Once a question is answered correctly, you will not be asked the same question again during the same session. When all questions are answered correctly the lesson will end. You will be given a score and returned to the title page of the Tutor program.

If you answer a question incorrectly, you will be asked the same question later during the same session, and again, until you finally get it correct.

When the question appears a second time, the correct answer will appear in a different position on the list of multiple choice responses.

It should be possible to leave the lesson at any time by selecting "Quit Lesson" from the pull down MENU. You will be given a score for what you have answered up to that point and then returned to the title page of the Tutor program.

### **Other Features of Editor**

Return to the Editor program, and open the Test lesson again. Try to add question 5. The Editor will inform you that you have already entered question 5 and asks if you want to change it. You will also be asked if you want to change the previous answer 5. Thus, you can change previous questions, answers, or question/answer pairs.

Choose "Add Question" and enter 11 at the blinking cursor. But suppose before you actually enter it you suddenly change your mind. Select "I changed my mind" from the pull down MENU



and you will be returned to the CAI Editor screen. The "I changed my mind" item is available from several places in the program. If you get stuck somewhere and are not sure what to do, check to see if this item is activated. As I mentioned earlier, you can delete questions in your lessons as you learn them, making your lessons more efficient. Let's try to delete question one. By the way, question one is the hardest to delete because to do that requires that the program copy question 2 to question 1, then question 3 to question 2, and so on until question 10 is copied to question 9. Try it. After it's done, make sure that if you ask to read question one it reads "This is question two." Similarly question two reads, "This is question three", and so on. The total number of questions reported at the top of the screen changes from ten to nine.

Now, let's check if the word wrap works. Choose "Add Question" and enter 10 at the blinking cursor. Type the following sentence exactly:

This is the first time that I enter a question into my wonderful compu

...and hold it right there. Look at the screen. You see one complete line of text and the blinking cursor has automatically jumped to the beginning of the next line. The last word on the first line is "compu." Now type the letter "t" and look at the screen. The partial word "compu" jumped to the second line and got joined with the "t" you just typed. This automatic word wrap will help you enter longer questions and answers without worrying if words get clipped at the end of a line. Verify that the wordwrap functions correctly at the end of the second line and at the end of the first two lines of the window as well.

The last function to verify is the "Delete Lesson File." Choose "Quit Lesson" and return to the title page of the Editor. Choose "Delete Lesson File" from the pull down MENU and enter "Test" at the blinking cursor. Good-bye Test! Choose "Delete Lesson File" again and the Directory of [This Disk] will show that "Test" is gone. You see that file called ".info"? That file is not really needed for proper operation of my program, and if you don't like its appearance, you may delete it now. If you want to keep the .info file there, return to the "CAI Editor" screen by choosing "I changed my mind" from the pull down MENU.

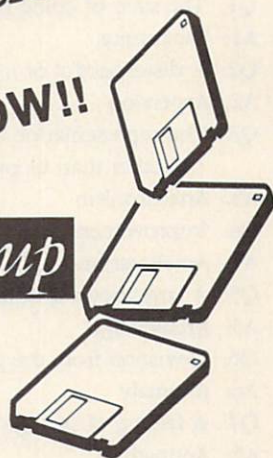
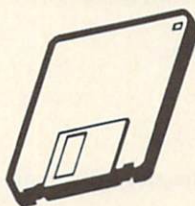
That should complete the debugging of your programs. Now you should change the END statements back to SYSTEM in both the Editor and Tutor programs, and make a back-up of this disk. You can overwrite your last backup disk if you have made changes or corrections since then. You have probably noticed that my Editor program saves your data to the same floppy disk that the programs are stored on. That's OK. There's enough room on one disk for more than 1000 question/answer pairs. Also, this simplifies the program and reduces the need for disk swaps. Just remember to keep an extra copy of the program package with no lessons saved to it as a master. As you fill up one disk with data, you can make a fresh copy from your master disk.

(continued)

BREAKTHROUGH

All Hard Drive Back-Up Programs were created equal . . . UNTIL NOW!!

**EZ-Backup**



100% better than any other hard drive back-up program  
EZ-Backup is a genuine breakthrough

**EZ-Backup actually manages the space on your backup disks--**

Before EZ-Backup the only way to get rid of the piles of incremental back-up disks that littered your desk was to do another full back up. This method is inconvenient and requires a whole new set of unformatted floppy disks.

With EZ-Backup you use the same set of disks for every incremental back-up. Only one full back-up required--Ever! Space on the disks is managed by deleting obsolete archive files and allowing you to save from 0-255 versions of each file. **Your files are saved--even if you have completely deleted them from the hard drive!**

**EZ-Backup comes with an optional warning screen--**

We all tend to put off doing backups. EZ-Backup's warning screen reminds you. If you would rather not be reminded--you have the option to shut the warning screen off.

**EZ-Backup prevents you from damaging valuable data--**

By checking the volume label, EZ-Backup keeps you from writing over important files.

**EZ-Backup uses Standard Amiga format--**

Files are archived in standard Amiga format and work with all standard utilities.

**EZ-Backup provides easy recovery of individual files--**

A simple to use mouse-oriented program allows you to recover individual files.

**Works with all Amiga-DOS compatible hard drives.  
(Amiga-DOS version 1.2 or higher)**

- \* Provides archive-bit utilities
- \* Multi-tasking
- \* Upgrades provided free for the first six months after program purchase!
- \* Free telephone support!
- \* Not copy protected
- \* Complete manual with examples

**Not more expensive just the best - \$49.95 - from:  
EZ-SOFT or an Amiga Dealer near you.**

21125 Chatsworth Street  
Chatsworth, CA 91311  
(818) 341-8681

**Dealer Inquiries Welcome**

**EZ-SOFT**



Now I think it would be a good idea to give you some simple examples of some data. Let me give you one in the subject I used most with this package, English Vocabulary. Please enter the following data:

- Q1: The state of being habitually silent.  
 A1: Taciturnity  
 Q2: A disrespectful or injurious remark or report.  
 A2: Aspersions  
 Q3: The representation of something as existing or occurring at other than its proper time.  
 A3: Anachronism  
 Q4: Improvement.  
 A4: Amelioration  
 Q5: A large body of water with many islands.  
 A5: Archipelago  
 Q6: Deviation from the general rule.  
 A6: Anomaly  
 Q7: A feeling of strong dislike.  
 A7: Antipathy  
 Q8: A readiness to act or serve; cheerful willingness.  
 A8: Alacrity  
 Q9: Angry dispute; a heated argument.  
 A9: Altercation  
 Q10: Evasion of the main point by indirect or roundabout speaking.  
 A10: Circumlocution

### Conclusion

I have tried in this article to get anyone, regardless of background, up and running with this program package. But I haven't said a single word about how the programs work. That may have disappointed some people. Maybe you would like to use the programs or even make modifications to them, but feel apprehensive because you don't know how they work. That's very legitimate. It is certainly easier to be told by the author how a program works rather than to fish through the code trying to figure out the algorithm yourself. I will gladly write another article explaining this entire program package fully, complete with suggestions for further development if there is some genuine interest. I would do this rather than try to conceal its operation behind compiled code, because I believe the future of fine products like the Amiga rests, in part, on the opportunity it provides for enthusiasts to learn more about computers (including programming). Magazine articles are an excellent way for us to share our work and contribute towards that goal. Write to Amazing Computing and let them know that you would like to see such an article written and I will willingly comply. If this program package helps you a little bit along the road to academic success, I will be happy.

[Next month, Part II will include the Editor program. The complete authoring system is available on Amazing on Disk, 3.9. See page 60 for more information -Ed.]

### Listing One: TUTOR

```
CAI.Program.1.5:

GOSUB Initialize.Program

Main.1:

GOSUB Title.Page

'activate menu - allow to choose:
' "New Lesson"
' "Quit Lesson"
' "GOTO Workbench"
GOSUB Clear.Keyboard
MENU 1,1,1
MENU 1,2,0
MENU 1,3,1
MENU 1,4,1
GOTO What.to.Do

Main.2:

GOTO Choose.Lesson

Main.3:

GOSUB Initialize.Lesson

Top.Main.Loop:

GOSUB Make.Current.Selection
GOSUB Find.Questions
GOSUB Juggle.Display
GOSUB Show.Selection
GOTO User.Response

Loop.5:
'disable all menu selections
MENU 1,1,0
MENU 1,2,0
MENU 1,3,0
MENU 1,4,0

IF D.Point%(User.R%)=1 THEN
  GOTO Question.Correct
ELSE
  GOTO Question.Wrong
END IF

Loop.6:

GOTO Top.Main.Loop

Make.Current.Selection:

ThisChoice%(1) = INT(RND*Quest.Left%)+1

FOR I%=2 TO 5
  Take.One:
  ThisChoice%(I%) =INT(RND*Total.Quest%)+1
  Duplicat% = 0
  FOR j=1 TO I%-1
    IF ThisChoice%(j)=ThisChoice%(I%) THEN Duplicat%+1
  NEXT j
  IF Duplicat%=1 THEN Take.One
NEXT I%
RETURN

Juggle.Display:

D.Point%(1)=INT(RND*5+1)
FOR I%=2 TO 5
  Pick.It:
  D.Point%(I%)=INT(RND*5+1)
  Duplicat%=0
  FOR j=1 TO I%-1
    IF D.Point%(j)=D.Point%(I%) THEN Duplicat%+1
  NEXT j
  IF Duplicat%=1 THEN Pick.It
NEXT I%
RETURN

Find.Questions:

GET #1, Report.Vector(ThisChoice%(1))
Show.Q% = Question$
Show.A$(1) = Answer$
```



```

FOR I%=2 TO 5
  GET #1, Report.Vector(ThisChoice%(I%))
  Show.A$(I%) = Answer$
NEXT I%

RETURN

Show.Selection:

IF MENU(0)<>0 THEN Show.Selection
IF MOUSE(0)<>0 THEN Show.Selection

IF First.Question=1 THEN
  First.Question=0
ELSE
  'IF MOUSE(0)<>0 THEN Show.Selection
  'enable Quit Lesson
  MENU 1,1,0
  MENU 1,2,1
  MENU 1,3,0
  MENU 1,4,0

  PRINT
  PRINT TAB(29) "Press Left mouse button"
  WHILE MOUSE(0)=0
    IF MENU(0)=1 THEN GOTO Project.Menu
  WEND
  WHILE MOUSE(0)<>0
    IF MENU(0)=1 THEN GOTO Project.Menu
  WEND
END IF

CLS
CALL Move$(WINDOW(8),40,11)
CALL Text$(WINDOW(8),SADD(Show.Q$),70)
CALL Move$(WINDOW(8),40,19)
CALL Text$(WINDOW(8),SADD(Show.Q$)+70,70)
CALL Move$(WINDOW(8),40,27)
CALL Text$(WINDOW(8),SADD(Show.Q$)+140,70)

CALL Move$(WINDOW(8),70,43)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(1))),60)
CALL Move$(WINDOW(8),70,51)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(1)))+60,60)
CALL Move$(WINDOW(8),70,59)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(1)))+120,60)

CALL Move$(WINDOW(8),70,75)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(2))),60)
CALL Move$(WINDOW(8),70,83)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(2)))+60,60)
CALL Move$(WINDOW(8),70,91)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(2)))+120,60)

CALL Move$(WINDOW(8),70,107)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(3))),60)
CALL Move$(WINDOW(8),70,115)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(3)))+60,60)
CALL Move$(WINDOW(8),70,123)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(3)))+120,60)

CALL Move$(WINDOW(8),70,139)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(4))),60)
CALL Move$(WINDOW(8),70,147)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(4)))+60,60)
CALL Move$(WINDOW(8),70,155)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(4)))+120,60)

CALL Move$(WINDOW(8),70,171)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(5))),60)
CALL Move$(WINDOW(8),70,179)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(5)))+60,60)
CALL Move$(WINDOW(8),70,187)
CALL Text$(WINDOW(8),SADD(Show.A$(D.Point%(5)))+120,60)

LINE (Q1.X1%,Q1.Y1%)-(Q1.X2%,Q1.Y2%),0,bf
LINE (Q2.X1%,Q2.Y1%)-(Q2.X2%,Q2.Y2%),0,bf
LINE (Q3.X1%,Q3.Y1%)-(Q3.X2%,Q3.Y2%),0,bf
LINE (Q4.X1%,Q4.Y1%)-(Q4.X2%,Q4.Y2%),0,bf
LINE (Q5.X1%,Q5.Y1%)-(Q5.X2%,Q5.Y2%),0,bf

LINE (Q1.X1%,Q1.Y1%)-(Q1.X2%,Q1.Y2%),,b
LINE (Q2.X1%,Q2.Y1%)-(Q2.X2%,Q2.Y2%),,b
LINE (Q3.X1%,Q3.Y1%)-(Q3.X2%,Q3.Y2%),,b
LINE (Q4.X1%,Q4.Y1%)-(Q4.X2%,Q4.Y2%),,b
LINE (Q5.X1%,Q5.Y1%)-(Q5.X2%,Q5.Y2%),,b

RETURN

User.Response:

'remove all mouse clicks
IF MOUSE(0)<>0 THEN GOTO User.Response
IF UCASE$(INKEY$)<>" " THEN GOTO User.Response

```

```

MENU 1,1,0
MENU 1,2,1
MENU 1,3,0
MENU 1,4,0

Ask.for.Answer:
WHILE MOUSE(0)=0
  IF UCASE$(INKEY$)="Q" THEN GOTO Terminate.2
  IF MENU(0)=1 THEN
    GOTO Project.Menu
  END IF
WEND

WHILE MOUSE(0)<>0
  IF MENU(0)=1 THEN
    GOTO Project.Menu
  END IF
WEND

IF MOUSE(3)<Q1.X1% OR MOUSE(3)>Q1.X2% THEN
  GOTO Ask.for.Answer
ELSEIF MOUSE(4)>Q1.Y1% AND MOUSE(4)<Q1.Y2% THEN
  LINE (Q1.X1%,Q1.Y1%)-(Q1.X2%,Q1.Y2%),3,bf:User.R%-1
ELSEIF MOUSE(4)>Q2.Y1% AND MOUSE(4)<Q2.Y2% THEN
  LINE (Q2.X1%,Q2.Y1%)-(Q2.X2%,Q2.Y2%),3,bf:User.R%-2
ELSEIF MOUSE(4)>Q3.Y1% AND MOUSE(4)<Q3.Y2% THEN
  LINE (Q3.X1%,Q3.Y1%)-(Q3.X2%,Q3.Y2%),3,bf:User.R%-3
ELSEIF MOUSE(4)>Q4.Y1% AND MOUSE(4)<Q4.Y2% THEN
  LINE (Q4.X1%,Q4.Y1%)-(Q4.X2%,Q4.Y2%),3,bf:User.R%-4
ELSEIF MOUSE(4)>Q5.Y1% AND MOUSE(4)<Q5.Y2% THEN
  LINE (Q5.X1%,Q5.Y1%)-(Q5.X2%,Q5.Y2%),3,bf:User.R%-5
ELSE
  GOTO Ask.for.Answer
END IF

GOTO Loop.5

Question.Correct:

SWAP Report.Vector(Quest.Left%), Report.Vector(ThisChoice%(1))
Quest.Left%=Quest.Left%-1
Number.of.Guesses=Number.of.Guesses+1

LINE (60,60)-(580,150),3,b
LINE (61,60)-(579,150),3,b
LINE (62,61)-(578,149),3,b
LINE (63,61)-(577,149),3,b
LINE (64,62)-(576,148),3,b
LINE (65,62)-(575,148),0,bf

CALL SetFont$(WINDOW(8),emerald20%)
CALL Move$(WINDOW(8),269,86)
PRINT "Correct"
CALL SetFont$(WINDOW(8),topaz11%)
CALL Move$(WINDOW(8),222,105)
PRINT "That's":Total.Quest%-Quest.Left%:"correct so far"
CALL Move$(WINDOW(8),242,120)
PRINT Quest.Left%:"questions left"
CALL SetFont$(WINDOW(8),topaz8%)

IF Quest.Left%=0 THEN
  Last.One.Done:
  IF MOUSE(0)<>0 THEN Last.One.Done
  PRINT
  PRINT TAB(29) "Press Left mouse button"
  WHILE MOUSE(0)=0:WEND
  WHILE MOUSE(0)<>0:WEND

  GOTO Terminate.2
END IF

GOTO Loop.6

Question.Wrong:

Number.of.Guesses=Number.of.Guesses+1

'LINE (60,60)-(580,150),2,bf
'LINE (65,62)-(575,148),0,bf
LINE (60,60)-(580,150),2,b
LINE (61,60)-(579,150),2,b
LINE (62,61)-(578,149),2,b
LINE (63,61)-(577,149),2,b
LINE (64,62)-(576,148),2,b
LINE (65,62)-(575,148),0,bf

CALL SetFont$(WINDOW(8),diamond20%)
CALL Move$(WINDOW(8),267,88)
COLOR 3,0
PRINT "WRONG !";
COLOR 1,0

```

(continued)



```

CALL SetFont$(WINDOW(8),topaz114)
Total.Wrong=Number.of.Guesses-(Total.Quest%-Quest.Left%)
CALL Move$(WINDOW(8),227,105)
PRINT "That's":Total.Wrong: "wrong so far!"
CALL Move$(WINDOW(8),180,120)
PRINT "I will ask you this question again"
CALL SetFont$(WINDOW(8),topaz84)

GOTO Loop.6

Choose.Lesson:

'start error trapping for disk
ON ERROR GOTO Error.Trap.1

CLS
LOCATE 2,1
message$="Listed below are Lessons on this disk"
PRINT TAB(FNT(80)) message$
PRINT
FILES "This Disk/"
FYPos%-CSRLIN

PRINT "                                Type in Lesson you want ":
MENU 1,1,0
MENU 1,3,1
MENU 1,4,1
Menu.Chosen%-0

GOSUB Input.Lesson.Name

IF Menu.Chosen%=1 THEN GOTO Project.Menu
MENU 1,3,0
MENU 1,4,0

LOCATE FYPos%,1

message$="..... Looking for "+Lesson.Name$+" ....."
PRINT TAB(FNT(80)) message$

OPEN "This Disk/"+Lesson.Name$ FOR INPUT AS #2
'INPUT #2,n
CLOSE #2

OPEN "This Disk/"+Lesson.Name$ AS #2 LEN=2
FIELD #2, 2 AS How.Many$
GET #2,1
n=CVI(How.Many$)

CLOSE #2

IF n<6 THEN
  request1$="Sorry...not enough questions in"
  request2$="this file. Add more from EDITOR"
  box1$="Pick Another"
  box2$="CANCEL"
  default%-1
  GOSUB AlertBox
  IF Answer%-1 THEN
    GOTO Choose.Lesson
  ELSE
    GOTO Main.1
  END IF
END IF

Total.Quest%-n
Quest.Left%-Total.Quest%

'Channel 1 for data on disk
OPEN "Lessons/"+Lesson.Name$ AS #1 LEN=390
FIELD #1, 210 AS Question$, 180 AS Answer$

'disable error trapping
ON ERROR GOTO 0

GOTO Main.3

Input.Lesson.Name:

GOSUB Clear.Keyboard

'initialize to null
Lesson.Name$=""

'what is row of last printed line
YPos%-CSRLIN
XPos%-46

'initialize PointC% to beginning of text
PointC%-1

Top.ILoop:
'Initialize Cursor
Cur$="- "

```

```

ILoop:
'Place cursor at current position
LOCATE YPos%,XPos%+XCur%(PointC%)
PRINT Cur$

'Change cursor so that it will blink
IF Cur$="- " THEN
  Cur$=" "
ELSEIF Cur$=" " THEN
  Cur$="- "
END IF

'initialize cursor timer
T%-0

'this is where keyboard input appears

KLoop:
a$=INKEY$
IF MENU(0)=1 THEN
  Menu.Chosen%-1
  GOTO Input.Done
END IF

'test if anything was entered
IF a$<>" " THEN Test

'nothing entered....increment cursor timer
T%-T%+1

'Test if cursor needs blinking
IF T%=40 THEN ILoop
GOTO KLoop

Test:
'test for carriage return
IF a$=CHR$(13) THEN Input.Done

'test for backspace key
IF a$=CHR$(8) THEN Back.Up

'test for maximum number of characters
IF PointC%-20 THEN KLoop

'test for illegal character
IF ASC(a$)<31 OR ASC(a$)>127 THEN KLoop

'input must be legitimate""

'echo input to screen
LOCATE YPos%,XPos%+XCur%(PointC%)
PRINT a$

'Store input in variable - concatenate
Lesson.Name$=Lesson.Name$+a$

'Increment PointC%
PointC%=PointC%+1

GOTO Top.ILoop

Back.Up:

'test if cursor is at beginning of text
IF PointC%=1 THEN KLoop

'erase cursor from present position
LOCATE YPos%,XPos%+XCur%(PointC%)

'decrement PointC%
PointC%=PointC%-1

'erase character from Lesson.Name$ variable
Lesson.Name$=LEFT$(Lesson.Name$,PointC%-1)

a$=""

GOTO ILoop

Input.Done:
RETURN

Initialize.Program:

DEF FNT(Z)=INT((Z-LEN(message$))/2)

SCREEN 1,640,320,2,2

'open full screen window - no title or gadgets
WINDOW 2,,(0,0)-(631,186),16, 1

LOCATE 10,1
message$="... Please Wait ..."

```



```

PRINT TAB(FNT(80)); message$

'define this programs menu choices
MENU 1,0,1," Stuff You Can Do "
MENU 1,1,0," Load Lesson <L> "
MENU 1,2,0," QUIT Lesson <Q> "
MENU 1,3,0," Goto Editor "
MENU 1,4,0," Goto Workbench "

'disable BASIC's menu choices
MENU 2,0,1,""
MENU 3,0,1,""
MENU 4,0,1,""

LIBRARY "graphics.library"
LIBRARY "diskfont.library"

DECLARE FUNCTION OpenDiskFont$( ) LIBRARY
DECLARE FUNCTION OpenFont$( ) LIBRARY

'fake a structure in BASIC
DIM TextAttr$(1)

'for topaz 8
TextAttr$(0)=SADD("topaz.font"+CHR$(0))
TextAttr$(1)=8*65536
topaz8=OpenFont$(VARPTR(TextAttr$(0)))
IF topaz8=0 THEN
  PRINT "I can't find topaz 8 font"
  FOR I%=1 TO 1000:NEXT I%
  GOTO Terminate.1
END IF

'for topaz 11
TextAttr$(0)=SADD("topaz.font"+CHR$(0))
TextAttr$(1)=11*65536
topaz11=OpenDiskFont$(VARPTR(TextAttr$(0)))
IF topaz11=0 THEN
  PRINT "I can't find topaz 11 font"
  FOR I%=1 TO 1000:NEXT I%
  GOTO Terminate.1
END IF

'for diamond 20
TextAttr$(0)=SADD("diamond.font"+CHR$(0))
TextAttr$(1)=20*65536
diamond20=OpenDiskFont$(VARPTR(TextAttr$(0)))
IF diamond20=0 THEN
  PRINT "I can't find diamond 20 font"
  FOR I%=1 TO 1000:NEXT I%
  GOTO Terminate.1
END IF

'for emerald 20
TextAttr$(0)=SADD("emerald.font"+CHR$(0))
TextAttr$(1)=20*65536
emerald20=OpenDiskFont$(VARPTR(TextAttr$(0)))
IF emerald20=0 THEN
  PRINT "I can't find emerald 20 font"
  FOR I%=1 TO 1000:NEXT I%
  GOTO Terminate.1
END IF

'arrays
DIM ThisChoice%(5)
DIM D.Point%(5)
DIM Show.A$(5)
DIM XCur%(20)

RANDOMIZE TIMER

'co-ordinates of question boxes
Q1.X1%=20:Q1.Y1%=38:Q1.X2%=35:Q1.Y2%=46
Q2.X1%=20:Q2.Y1%=70:Q2.X2%=35:Q2.Y2%=78
Q3.X1%=20:Q3.Y1%=102:Q3.X2%=35:Q3.Y2%=110
Q4.X1%=20:Q4.Y1%=134:Q4.X2%=35:Q4.Y2%=142
Q5.X1%=20:Q5.Y1%=166:Q5.X2%=35:Q5.Y2%=174

'flag for initialization
First.Time.Thru%=1

FOR I%=1 TO 20
  XCur%(I%)=I%
NEXT I%

RETURN

Initialize.Lesson:

IF First.Time.Thru% = 0 THEN
  ERASE Report.Vector, Mark.Vector
ELSE
  First.Time.Thru% = 0
END IF
First.Question=1

```

```

DIM Report.Vector(n)
DIM Mark.Vector(n)

Number.of.Guesses = 0

FOR I%=1 TO n
  Report.Vector(I%)=I%
  Mark.Vector(I%)=0
NEXT I%

RETURN

Title.Page:

CLS
LINE (175,13)-(445,55),1,b
LINE (174,14)-(174,56),2
LINE (173,15)-(173,57),2
LINE -(443,57),2
LINE (171,15)-(171,57),2
LINE -(441,57),2
CALL SetFont$(WINDOW(8),topaz11)
CALL Move$(WINDOW(8),208,27)
PRINT "Computer Aided Instruction"
CALL SetFont$(WINDOW(8),diamond20)
CALL Move$(WINDOW(8),270,110)
message$="TUTOR"
PRINT message$
CALL SetFont$(WINDOW(8),topaz8)
CALL Move$(WINDOW(8),263,47)
PRINT "Revision 1.5"
CALL Move$(WINDOW(8),239,75)
PRINT "by Paul Castonguay"

CALL Move$(WINDOW(8),35,130)
PRINT "This program allows you to study material";
PRINT "that you have previously"
CALL Move$(WINDOW(8),35,140)
PRINT "entered using the EDITOR in this package."
COLOR 3,0
CALL Move$(WINDOW(8),192,179)
PRINT "Press RIGHT mouse button and"
LOCATE 24,26
PRINT "Select from PULL DOWN MENU";
COLOR 1,0

RETURN

Clear.Keyboard:
IF INKEY$<>" " THEN Clear.Keyboard
IF MENU(0)<>0 THEN Clear.Keyboard
RETURN

What.to.Do:
GOSUB Clear.Keyboard
What.Loop:
'wait for user to decide what to do
IF UCASE$(INKEY$)="L" THEN GOTO Main.2
IF MENU(0)=1 THEN
  MENU 1,1,0
  MENU 1,2,0
  MENU 1,3,0
  MENU 1,4,0
  GOTO Project.Menu
END IF
GOTO What.Loop

Project.Menu:

IF MENU(1) = 1 THEN GOTO Main.2

IF MENU(1) = 2 THEN GOTO Terminate.2

IF MENU(1) = 3 THEN Load.Editor

IF MENU(1) = 4 THEN Terminate.1

GOTO What.to.Do

Terminate:
Terminate.1:

'replace memory used by window #2
WINDOW CLOSE 2
SCREEN CLOSE 1
CLS
're-establish BASIC's menu choices
MENU RESET
LIBRARY CLOSE
END

```

(continued)



**ATTENTION****READERS**

# AC NEEDS YOU !

Amazing Computing™ cannot determine the dependability of advertisers from their advertisements alone. We need your feedback. If you have a problem with an advertiser in AC™, please send a complete description of the incident, in writing to:

**Ad Complaints**  
**PIM Publications, Inc.**  
**Amazing Computing**  
**P.O. Box 869**  
**Fall River, MA 02722**

Be sure to include any correspondence you have had with the advertiser, along with the names of the individuals involved. Your assistance is greatly appreciated.

Load.Editor:

```
'replace memory used by window #2
WINDOW CLOSE 2
SCREEN CLOSE 1
CLS
're-establish BASIC's menu choices
MENU RESET
LIBRARY CLOSE
RUN "Editor"
```

Terminate.2:

```
CLOSE #1
MENU 1,2,0
CLS
LOCATE 10,1
PRINT TAB(18) "It took you";Number.of.Guesses;
PRINT "guesses to answer";Total.Quest%-Quest.Left%:"questions!"
COLOR 3,0
LOCATE 24,1
message$=" Press LEFT mouse button or <RETURN> "
PRINT TAB(FNT(80)) message$;
COLOR 1,0
```

```
Wait.Read.Score.1:
IF MOUSE(0)<>0 THEN Wait.Read.Score.1
WHILE MOUSE(0)=0
IF INKEY$=CHR$(13) THEN GOTO Main.1
WEND
WHILE MOUSE(0)<>0
IF INKEY$=CHR$(13) THEN GOTO Main.1
WEND
GOTO Main.1
```

Error.Trap.1:

```
BEEP
WINDOW 2

IF ERR=53 THEN
request1$="I can't find "+Lesson.Name$
request2$="Did you mis-spelled it?"
GOTO ExitError1
END IF
```

```
IF ERR=64 THEN
request1$="BAD FILENAME."
request2$=""
GOTO ExitError2
END IF

IF ERR=68 THEN
request1$="DEVICE UNAVAILABLE."
request2$=""
GOTO ExitError2
END IF
```

```
request1$="ERROR NUMBER"+STR$(ERR)
request2$=""
GOTO ExitError2
```

```
ExitError1:
box1$="retry"
box2$="CANCEL"
default%=1
GOSUB AlertBox
IF Answer%=1 THEN
CLOSE #2
RESUME Choose.Lesson
ELSE
CLOSE #2
RESUME Main.1
END IF
```

```
ExitError2:
box1$="retry"
box2$="CANCEL"
default%=2
GOSUB AlertBox
IF Answer%=1 THEN
CLOSE #2
RESUME Choose.Lesson
ELSE
CLOSE #2
RESUME Main.1
END IF
```

AlertBox:

```
WINDOW 3,"Program Request",(0,0)-(311,45),16,1
```

```
PRINT LEFT$(request1$,39)
PRINT LEFT$(request2$,39)
```

```
b1$=LEFT$(box1$,12)
b2$=LEFT$(box2$,12)
```

```
boxsize1=(LEN(b1$)+2)*10
boxsize2=(LEN(b2$)+2)*10
```

```
x1=(312-(boxsize1+boxsize2))/3
x2=x1+boxsize1
x3=x1+x2
x4=x3+boxsize2
```

```
LINE(x1,20)-(x2,38),2,b
LINE(x3,20)-(x4,38),2,b
```

```
IF default%=1 THEN LINE(x1+2,22)-(x2-2,36),3,b
IF default%=2 THEN LINE(x3+2,22)-(x4-2,36),3,b
```

```
LOCATE 4,1
PRINT PTAB(x1+10);b1$;
PRINT PTAB(x3+10);b2$
```

Reqloop:

```
WHILE MOUSE(0)=0:WEND
```

```
m1=MOUSE(1)
m2=MOUSE(2)
```

```
IF m1>x1 AND m1<x2 AND m2>20 AND m2<38 THEN
Answer%=1
LINE(x1,20)-(x2,38),1,bf
```

```
ELSEIF m1>x3 AND m1<x4 AND m2>20 AND m2<38 THEN
Answer%=0
LINE(x3,20)-(x4,38),1,bf
```

```
ELSE
GOTO Reqloop
END IF
```

```
WHILE MOUSE(0)<>0:WEND
WINDOW CLOSE 3
RETURN
```

•AC•



# GELS IN MULTI-FORTH

## PART TWO: SCREENPLAY

by John Bushakra

Only on rare occasions can a programmer finish a project, and have that project stay finished. There always seems to be some little quirk that needs de-quirking, fine tuning, or in this case, overhauling.

Case in point: the IFF converter program (affectionately known in my household as "The Program That Wouldn't Die") which appeared in this magazine under the title "Gels In Multi-Forth." You can imagine my shock, when, as I lay blissfully sleeping one night, I thought I was rudely awakened by my troubled program.

"John," it inquired, "Might we speak a moment?"

"Er," I replied, stymied and at a loss for words, never having spoken to a program before. At least not in words containing more than four letters...

"I'm too hard to use," it muttered. "I don't think people will like me."

"That's silly," I countered. "You're very useful. Now go back to sleep. Or whatever." I pulled the covers over my head with a sharp jerk. (This had the effect of uncovering my wife, who testily mumbled something about me sleeping the rest of the night with a disk drive, if I didn't be quiet and go back to sleep.)

"But what if the user has a bunch of brushes to convert, and can't remember any of the file names? I need a file requester! It'd be neat if I had my very own screen and window to run in, with all the Gel variables represented by gadgets and stuff," the program proposed.

"Umph!" I said. "Do you know how many gadgets that is?"

"Thirty-six," answered the program.

"Holy BeeGeeBees, do you know how many relocatable addresses that comes to?" I whined.

"Seventy-two, not counting IntuiTexts and Borders," countered the program.

"Oh, thank you. I suppose you want a menu too..."

"Of course! Well, just a small one will do," quipped the program.

"Harumph," I told the program. "Okay, I'll start on it tomorrow. Um, is that it?"

"I can't think of anything else right now, except your disk drive heads are very dirty. If you don't clean them soon, you're going to lose every single program you own," warned the program. With that light-hearted announcement, the BrushConverter program bounced happily back to its box on my desk.

Needless to say, the entire Converter program has been "Intuitioned," and the format of the files it creates completely changed. Now it stores all views of an animation sequence, including ColorMap information, in one file. I hope the revised Converter will soon be living on an Amicus Disk. Watch for it; it's a vast improvement, and a good example of using Intuition from Multi-Forth.

Whatever the case may be, when this show is over, we will have developed tools enabling us to do the following:

- 1) Use the power and versatility of an art program, such as Deluxe Paint, to draw IFF brushes.
- 2) Convert the brushes to raw bitmap form, with pertinent Gel information thrown in. This information will be stored in a file.
- 3) Easily put up any type of screens on which to display the Gels.
- 4) Finally, we'll have a group of functions, which allow us, at the top level, to pass them only the name of a Gelfile, plus a few other miscellaneous flags and variables. These functions will handle all the dirty work associated with Gels (and there is a lot): Allocating memory, initializing flags, linking together, the whole shot.

(continued)



As all of us have come to find out, the Amiga is a wonderfully versatile machine. It allows the programmer seemingly endless options in setting up a program. Unfortunately (there's always a catch!), the magic equation that holds here is:

Versatility = (Options \* Headaches)^2.

Those of you who just started programming this beast, take heed! It behooves you to build a collection of tools to do such mundane things as opening screens and allocating bit maps. What's so hard about opening screens? Nothing—but that's just the point; it's boring! It would be nice to focus on more important aspects of your program, without wasting a lot of time setting this stuff up.

Now, we want to look at general purpose routines and structures to open any type of Intuition screen.

Listing One is the file I named screens.misc. The first thing you see in it is a structure called bmapack. Although there is nothing fancy about it, it provides us with the first lesson of "tool building": CONSOLIDATE! Try to keep as much information as possible under one roof. The members of this structure are simply variables associated with BitMaps: width, height, depth, and finally, the BitMap structure itself (NOT a pointer to a BitMap). The width and height of a bit map are independent of the width and height of the ViewPort.

The rest of the functions deal with allocating display memory, freeing display memory, reading color map files (produced with the OLD version of my Converter program), and opening and closing screens.

All of the functions utilize local variables to generalize their respective procedures. Generally, in Forth, all variables are global (that is, they can be seen by any word). But why limit ourselves to using a specific variable name? Someday we'll want to make a double buffered display, or a dual playfield, and we'll need more than one bit map. For instance, the makebitmap function would be useless to us if it only operated on a single, globally declared bit map!

The readcmap function expects to find a file containing color data. The data is of the form 0xRGB (sorry for the C Hex notation, Forth people), a 16 bit word containing Red, Green, and Blue intensity values. Remember, color data in an IFF file is stored (in Hex) as F0 F0F0 (white, for example). This is stored as 0FFF in my files. The function uses the value of the local constant l#cregs to determine how many color registers to read from the file. The value is multiplied by two, since the Read function expects a byte count on the stack (e.g. 32 color registers = 32 \* 2 = 64 bytes to read in). Also, remember that dosRead is the AmigaDOS Read, not Multi-Forth's.

The function "onscreen" opens both CUSTOMBITMAP and non-CUSTOMBITMAP screens. Notice that it checks to see if you

sent it a bmapack before trying to create a custom bit map. If you sent it a file name, it then calls readcmap.

Offscreen simply closes the screen, and deallocates display memory.

Listing Two contains a structure and two functions for dealing with dual playfields. The dualscrn structure is merely a consolidation of variables we need to associate with the second playfield (Intuition takes care of the first). To create a dual playfield display, you need to allocate another RasInfo, BitMap, and Rastport structure, in addition to keeping track of the dimensions of the new playfield (the two playfields need not be the same width, height, or even the same depth).

The official word from the Higher Ups at Commodore is that setting the DUALPF bit of the ViewModes variable in your NewScreen structure is not the best way to get a dual playfield display. So we get to perform some tomfoolery here, which is even more reason to factor this out into a separate function and forget about it.

The function "makeitdual" does the dirty work for us. It is intended to be called immediately after onscreen. We need to allocate more display memory, then store the newBitMap in our number two RastPort. Then, while Intuition isn't looking, we have to link our second RasInfo to the one Intuition gave us when we opened the screen. Now we set the DUALPF bit. Calls to MakeScreen and ReThinkDisplay put our clandestine changes into effect.

Doing things this way keeps Intuition from using your second playfield for rendering gadgets, menus and all the other stuff Intuition is famous for. The catch is, you've got to remove the evidence before you call CloseScreen, hence the function "freedual." To cover our tracks, we need to store a zero in the link field of the first RasInfo (the one Intuition gave us). Then, we turn off the DUALPF bit with an XOR (exclusive OR) operation. Next, call MakeScreen and ReThinkDisplay, and our screen is back to a plain single playfield. All that is left to do is free up the display memory for the second BitMap. Don't forget, you need to call this function before you call offscreen.

When using dual playfield displays, the system uses only the lower sixteen color registers, and those registers are split into two halves. That is, playfield one uses registers zero through seven, and playfield two uses registers eight through fifteen.

The following table summarizes playfield depth to color register correspondence. It's in the ROM

Kernel Manual, but I'll produce

it here to minimize cross referencing.

PF 1 Depth	Color Regs.	PF 2 Depth	Color Regs.
1	0-1	1	8-9
2	0-3	1	8-9
2	0-3	2	8-11
3	0-7	2	8-11
3	0-7	3	8-15



Now to Listing Three, which contains code to put up and toggle the BitMaps of a double buffered display. Double buffering is a technique used to produce smooth animation. You create two bit maps, drawing into one while displaying the other. When the display is completely updated, switch frames to show the updated display. It's tricky, and very expensive memory-wise. A double buffered, 640 x 200 display, four planes deep, will cost you 128,000 bytes. If you've got expansion RAM, go for it. The effects are well worth the cost.

Creating the display is easy. First, we need to gather together all variables associated with double buffering, and put them all in one structure. Next, I've declared a global variable called *frametog*, which decides which bit plane to draw in, and which to display. The user need never be concerned with this variable. Just say *togglemaps* in a definition somewhere, and the work is done. The definition of *togglemaps* just checks the state of *frametog*. If it's non-zero, we draw into BitMap one; otherwise, draw into BitMap zero. Notice the BitMap that gets displayed is the one stored in the *RasInfo* structure of our screen. After the switch of BitMaps, we have to call *MakeScreen* and *ReThinkDisplay* to put the change into effect.

The definitions of *makedbuf* and *offdbuf* are similar to *onscreen* and *offscreen*. *Makedbuf* allocates the display memory, and opens the screen with BitMap zero. It then allocates memory for the second bit map, and toggles the frames, to start us off drawing into bit map zero. *Offdbuf* simply closes the screen and deallocates display memory.

These functions replace those in the screens .miscfile. So, when you use the double buffering words, the only functions you need from that file are *makebitmap*, *freebitmap*, and *readmap*. Since it is a little wasteful to include functions you don't need, you might consider importing those functions to *dblstruct.f*. This way you wouldn't have to include *screens.misc*.

Now let's stop for a moment and consider the big picture. It's likely you might want to use some Gels in your double buffered display. If that is the case, you should take out the calls to *MakeScreen* and *ReThinkDisplay* in the function *togglemaps*. Since some Gels, like *VirtualSprites*, require you to call *MrgCop* and *LoadView* after each update, it would be wasteful, and seriously degrading to performance, if you called those functions twice (*MakeScreen* and *ReThinkDisplay* are *Intuition*'s version of *MrgCop* and *LoadView*). Try to arrange your code so you update your entire display, including any Gels you might have. Then toggle the bit maps, and finally, call *MakeScreen* and *ReThinkDisplay* just once. All your changes will be put into effect at the same time.

Now we need to define global constants and structures to pass to our functions. The next three listings show examples of how we might want to build these files. Listing Four is for normal (single playfield) displays; Five is a dual playfield example; and Six is for double buffering.

First, decide what type of display your application needs, then go to the appropriate file, change the constants to suit your needs, and include that file in your program. Each of the three files automatically includes the support files it needs. For instance, the dual playfield example loads the file *dualstruct.f*. To open your screen, simply pass your global structures along to the functions, and the dirty work is done. No fuss, no muss, and best of all, very little initialization.

Now, on to something slightly more interesting: joystick and input device handlers. My first interest in the input device came from my desire to attach a joystick to either controller port. It seemed silly to be able to use only the right controller. As I explored the input device via the *ROM Kernel Manual*, I became more interested. The device merges events from the timer, keyboard, and either controller port into a single input event stream.

"That's perfect!" I thought. "Now, how do I get in on this?"

The answer is a combination of "be extremely careful" (for the keyboard), and "you really shouldn't" (for the controller port). What I gathered from reading the *RKM*, along with a little experimentation, is that the input device is opened when you boot the machine, and at least two event handlers are installed in a chain of handlers. These two are the king and queen of event handling: *AmigaDOS* (with its console), and *Intuition* (with its mouse and gadget arbitration). Briefly, what happens goes like this:

The input device constantly monitors the keyboard, mouse, and timer. Certain conditions are pre-set to trigger an input event, namely mouse movement, button transitions, and key strokes.

Events are taken care of by handlers, which are linked in a chain, according to a certain priority. Higher numbers have higher priority; *Intuition*, for instance, supposedly has a priority of fifty. If you want any say over how events are handled, you have to write code to check for events you're interested in, and install your code in the chain.

Now, when an input event is triggered, all handlers are called in order of priority. Events come in a structure called, surprisingly, an *InputEvent*, and several events may be linked together. Each handler can remove or modify the chain of events that pass through it. It can also add new ones to the list. When the handler is done, it passes the address of the first event in the new chain in data register D0. Event handling then proceeds to the next handler in the chain, and on down the line, until someone returns a zero in D0. If you want to have about two seconds of fun one afternoon, try to write a little handler, install it in front of *Intuition* (priority = 51), and when it's finished, pass a zero in register D0. What you are doing is signalling everyone else in the chain that there are no more events to process. So guess what? Nothing gets done. No keyboard, no

(continued)



mouse, no nasty disk grinding... We can see the implications here. If you were so inclined, you could write your own event handler, pass a zero to the other handlers, and effectively have the whole machine to yourself. But that sort of thinking is reserved for the days of eight-bit hacking. "We're above that sort of thing now," as W.C. Fields once said.

It is possible to change the conditions triggering an input event. It is also possible to change the controller port the device monitors, as well as the type of controller connected to that port. Alas, in the course of my experiments, I discovered that such things are best left alone. Changing the port the mouse is at is pretty simple, and probably harmless, but changing the controller type is equivalent to taking over the machine, and that's a practice which is sure to result in your being cast into the dreaded Amiga Programmer's Abyss. Even if you were to change the controller type to a joystick, for example, there is still no way for you to gather movement events easily, as the input device lacks a read command. Notice I said "gather easily." One could write a handler and install it in the chain, but the relative merits of this are few. At this point, it is easier to just grit your teeth and open the gameport device, which is what I ended up doing.

Handling keyboard events leads to a similar situation. But here you're not faced with the scary prospect of taking over the machine. Why would one consider monitoring the keyboard without using the console device? Suppose you are designing a High Speed Graphics-Oriented Application (a video game), and you need keyboard events. If possible, you would like to avoid opening a window; it seems unnecessary. The console device is a big hassle, especially if you're not interested in the line editing features it provides. Having Intuition send you RAWKEY events would be ideal, except Intuition won't send you anything unless you open a window. (I still don't understand why there is no option to attach an IDCMP to a screen.)

So both of those ideas are out. There is a nice chapter on the keyboard device in the *RKM*, but when you get to the program example at the end, you are cautioned not to fool with the keyboard directly, unless you want to take over the machine. The only thing left to do is install your own input device handler for catching keyboard events. Here, the overhead isn't great (except for loading the assembler), and we can do this without taking over the machine.

So far, I've talked a lot about the input device without mentioning any details of working with it. Let's do that now. Actually, communicating with the input device is pleasantly simple. If you're using Forth, you can do it interactively. Listing Seven contains the necessary ingredients for communicating with the gameport and input devices.

The devicepack structure is (what else?) a gathering of common structures needed in device communication. Opening the devices is simple. All you need to do is pass the name of the

device, the desired unit (the input device is always unit zero; the right controller port is unit one for the gameport device), an *IOStdReq* structure (some devices require a special form of this structure), and a fourth mystery variable, which the *RKM* explains might be used to request exclusive access to a device (just send a zero, and everyone will be happy). Since this is a standard process, I've factored it out into a separate function called *setupdevice*.

Commands sent to the two devices also follow the same formats. Specific device commands can be found in the devices directory in the language of your choice (in Multi-Forth it's *devices/gameport.f* for the gameport, and *devices/input.f* for the input device). You first send the specific command (e.g. *IND\_ADDHANDLER*, to add a handler to the chain), the length of the data structure the device needs to access, the address of that data, and finally, the *IOStdReq* you used to open the device. Since this too is standard for both devices, I've put it in its own function, called *docommand*.

There are two functions we want to look at next. One installs the handler (*setupkbd*), and the other is the handler itself. To install a handler, we also need an Interrupt structure. In this structure, we store the priority (at 51) and the address of the first instruction in our handler. To get the address of your handler, you simply do this:

```
`your-handler-name 4-
```

The ' ("tick") finds the parameter field address in the dictionary, and the 4- is needed to get back to the address of the code. We also need to store an address in the Interrupt's *is\_data* field. This address will be passed to our handler, and the handler may use it any way it wants to (subject, of course, to restrictions of common sense and consistency). Notice the address I store is actually an address of a pointer variable in the devicepack structure. The address of each event chain our handler sees will be stored in this address. Next, we send an *IND\_ADDHANDLER* off to *docommand*, along with the length of an Interrupt structure, the Interrupt structure itself, and our *IOStdReq*. Upon return from *DoIO*, our handler will be in the chain, sending us addresses of *InputEvents*.

The handler itself is wonderfully simple. In its three lines, you'll find the sum total of my knowledge of 68000 assembly language. (Not really. I also know all about the GURU instruction, which jumps randomly to code that draws a red box and a lot of meaningless text, then hangs up the machine.) When your handler is called by the system, the address of the current *InputEvent* is in A0, and your data area address is in A1. So, all we want to do is move the address at A0 to the address contained in A1. We then pass on what's unchanged in A0 to the next handler, Intuition.

Now that we've got an event, or a whole chain of them, we can process. Check out the *InputEvent* structure in the file of the same name in the devices directory, and you'll find members



telling you which class of event occurred; and, if a key was pressed, which key it was. (You might also notice that an InputEvent and an IntuiMessage structure look suspiciously alike.)

Now you can do anything you want with the event, but you'd better do it fast: new events come in all the time. I've written a little function called `monitorkbd`, which is an example of what you might want to do. It's just a while loop which goes until it finds no more events in the chain our handler gave us. In the loop, you could execute whatever code you want. I put a little something in there to flash the screen, just to see if it was working.

When you're finished using the handler, you remove it from the chain, and close the input device. The function "takedownkbd" does this for us. Removing a handler is similar to adding one; you send the same data structures to the `docommand` function.

Opening the gameport device is similar to opening the input device. The function "setupstick" performs the job, sets the controller type to be joystick, and sets the conditions which trigger an event at the gameport. To set trigger conditions, fill out a `GamePortTrigger` structure. You are responsible for deciding how you want the buttons reported (button up, down, or both), how far the controller has to move to generate an event (pertinent for a mouse; joysticks use a 1 here). Finally, you must state how much time should elapse between reports, if the other two conditions are not met.

In my example, I asked for reports when the button was either up or down, with a timeout value of one. This means the gameport device will send me an event every sixtieth of a second, regardless of whether or not it was actually moved (the timeout value is specified in 1/60th second increments). If you were to set a zero here, you would only receive reports when the controller were moved to a new position.

To read events from the controller port, you send a `GPD_READEVENT` command to the device. The data area you send is an `InputEvent` structure. The device fills the structure with information on X, Y, and button transitions. Notice that the `docommand` uses `DoIO` to read the event, which doesn't return until the request is satisfied. This way, we don't have to worry about sending the request, then waiting around at some port for the answer (`SendIO` requires this). A joystick report every 1/60th of a second is "close enough for jazz," as they say.

When finished with the gameport, you need to close it, and deallocate your message port. For safety's sake you can also set the controller type to `GPT_NOCONTROLLER` before you exit. In other words, leave the port exactly the way you found it. If you do, you'll be accepted into the ranks of that illustrious group, the Divine Order of Cooperative Amiga Programmers.

Listing Eight shows how to use the single playfield, input device handler, and gameport initialization routines. A screen is opened and some random junk is drawn on it, so we'll have something to scroll around in.

The simple way of scrolling is to change the values of `RX` and `RY` `Offset` in the `RasInfo` structure. These values represent the offset of the bit map from the viewport. To scroll to the right, you increment `RX`. To scroll down, increment `RY`. After changing the values, call `ScrollVPort`, then `MakeScreen` and `ReThinkDisplay` to effect the changes.

This way of scrolling is handy indeed, except it doesn't solve the problem of creating a scrolling display that wraps around. That is, after you scroll the width or height of your bit map, you want to re-enter it from the other side. To do this, you would need to check if you scrolled the width of the bit map. If you did, reset the `RxOffset` to zero. If you go off the left edge (`RX` becoming negative), you need to start over by subtracting your scroll increment from the total width of the bitmap, and storing the result as the new `RXOffset`. For example, if you were scrolling left (negative increments) in increments of two pixels, and reached an `RX` of zero, subtract two from your bit map width (600 in my example), leaving an `RXOffset` of 598.

That trick works nicely, but only in the X direction. If you try it on the Y axis, your HAM PacMan will wander out of his maze and into (depending on how many bit planes you've selected for your display) some of the most colorful and exotic varieties of screen garbage available on any microcomputer. So unless there is some trick I was unable to discover, using `ScrollVPort` doesn't completely solve the wrap around scrolling display problem. If any bright people out there have any ideas, I'd love to hear them (and so would a lot of other people, probably).

So that (finally) wraps up this installment on opening screens and devices. Next time, we'll be displaying Gels, I promise.

[All eight listings are available on *Amazing on Disk*, 3.9. See page 60 for more information. -Ed.]

### Listing One: screens.misc

```
\ -> Listing One: screens.misc
\ -> Miscellaneous routines for putting up Amiga displays.
\ -> These programs were created entirely with Amiga Multi-Forth,
\ -> a product of: Creative Solutions
\ -> 4701 Randolph road
\ -> Suite 12
\ -> Rockville, MD. 20852
\ ->

\ The bmapack struct consolidates variables associated with, well,
\ with BitMaps...

structure bmapack
    SHORT: +bpWidth ( The width of the BitMap, in pixels. )
    SHORT: +bpHeight ( The Height, or # of rows. )
    SHORT: +bpDepth ( The number of bitplanes... )
    BitMap STRUCT: +bpBitMap ( And the structure itself. Note this )
                                ( is NOT a pointer to a BitMap struct. )
structure.end
```

(continued)



```

\
\ In the bitmap structure is an array of pointers to bitplanes.
\ 'planeindex' indexes into that array, given the desired plane
\ number on the stack.
: planeindex ( plane index num - addr. )
    4 * + ;

\
\ makebitmap:
\ Synopsis: Responsible for allocating display memory for each
\ bitplane in the user's bitmap.
\
\ Inputs: Width, Height, depth, and 'map', the address of a
\ BitMap struct. Width height & depth are integers.
\ Returns: The address of the BitMap struct (passed to this
\ function), or aborts on memory allocation error.
\
\ The local var. 'rassize' is the size in bytes of one plane of
\ the display. The formula is: (Width / 8) * Height.
\ The local var. 'firstword' is the address of the first word of
\ display memory; this address is returned by AllocRaster.
: makebitmap ( width / height / depth / bitmap - bitmap )
    0 0 locals| rassize firstword map ldepth lheight lwidth |
    lwidth 8/ lheight * to rassize
    map ldepth lwidth lheight InitBitMap
    ldepth 0 do
        lwidth lheight AllocRaster to firstword
        firstword 0=
        if
            ?turnkey if bye else abort then
        else
            firstword rassize 1 BltClear
            firstword map +bmPlanes i planeindex |
            then
        loop
    map ;

\
\ freebitmap:
\ Synopsis: Frees up memory allocated with makebitmap, above.
\ Inputs: Width, Height, depth (integers), and the address
\ of a bitmap struct.
\ Returns: Nothing.
: freebitmap ( width / height / depth / bitmap - )
    locals| map ldepth lheight lwidth |
    ldepth 0 do
        map +bmPlanes i planeindex @
        lwidth lheight FreeRaster
    loop ;

\
\ readcmap:
\ Synopsis: Reads in RGB color values into the supplied array.
\ Inputs: 1) Filename: Null terminated name of the cmap file.
\ 2) #cregs : Number of color registers in this cmap.
\ 3) ctable : Address of the array in which to store the
\ colors.
\ Returns: ctable, the address of an initialized colortable.
: readcmap ( filename / #cregs / ctable - ctable )
    locals| ctable l#cregs filename |
    filename MODE_OLDFILE dosOpen dup
    if
        dup ( fileid ) ctable l#cregs 2* dosRead drop
    else
        drop
        ?turnkey if bye else abort then ( no return from here... )
    then
    ( fileid ) dosClose

    ctable ;

\
\ onscreen:
\ Synopsis: Calls OpenScreen, to put up the required display.
\ Inputs : 1) NewScreen: Address of a NewScreen struct.
\ 2) bmapack : Address of an initialized bmapack struct
\ (see above for the definition).
\ Send a zero in this place to indicate no
\ Custom BitMap desired.
\ 3) cmapfile: Null terminated name of the colormap
\ file. Send a zero in this place to
\ indicate no file to be read.
\ 4) #cregs : Number of color registers in the cmap.
\ 5) colortable: Address of an array to hold color
\ values.
\ Returns: Address of the opened screen.
: onscreen
    ( newscreen / bmapack / cmapfile / #cregs / colortable -
    screen addr )

```

```

locals| ctable l#cregs cmapname bmap lnewscrn |

bmap if
    bmap +bpWidth w@
    bmap +bpHeight w@
    bmap +bpDepth w@
    bmap +bpBitMap makebitmap
    lnewscrn +nsCustomBitMap !
then

lnewscrn OpenScreen

cmapname if
    cmapname l#cregs ctable readcmap
    currentscreen @ +scViewport swap ( vp ctable )
    l#cregs LoadRGB4
    then

currentscreen @ ;

\
\ offscreen
\ Synopsis: Closes a custom screen, frees display memory.
\ Inputs : 1) Screen : Address of an open screen.
\ 2) bmapack: if this screen had a custom bitmap,
\ send its address here, and the memory
\ for the bitmap will be freed. Send
\ a 0 if you didn't use a custom bitmap.
\ Returns: Nothing.
: offscreen ( screen / bmapack )
    locals| bmap lscreen |

    lscreen CloseScreen

    bmap if
        bmap +bpWidth w@
        bmap +bpHeight w@
        bmap +bpDepth w@
        bmap +bpBitMap freebitmap
    then ;

```

## Listing Two: dualstruct.f

```

\ Listing Two: dualstruct.f
\ Structures and functions for creating Dual Playfield
\ displays.
\
\ dualscrn is a consolidation of variables associated with
\ dual playfields.

structure dualscrn
    RasInfo STRUCT: +dsRasInfo2 ( second rasinfo )
    BitMap STRUCT: +dsBitMap2 ( second bitmap )
    RastPort STRUCT: +dsRastPort2 ( second rastport )
    SHORT: +dsWidth ( width of the BitMap )
    SHORT: +dsHeight ( Height of the BitMap )
    SHORT: +dsDepth ( Depth of the BitMap )
structure.end

\
\ makeitdual:
\ Synopsis: Creates a dual playfield from an opened intuition
\ custom screen.
\ Inputs: 1) screen: Address of an opened screen.
\ 2) dualstruct: Address of a dualscrn structure.
\ Returns: Address of the screen.
: makeitdual ( screen dualstruct - screen addr )
    locals| dual lscreen |

    dual +dsRastPort2 InitRastPort ( do initialization...)
    dual +dsWidth w@
    dual +dsHeight w@
    dual +dsDepth w@
    dual +dsBitMap2 makebitmap ( make the BitMap... )
    dual +dsRastPort2 +rpBitMap ! ( Store BitMap address )
    ( into the appropriate )
    ( structures.... )

    dual +dsBitMap2
    dual +dsRasInfo2 +riBitMap !

    dual +dsRasInfo2 ( Link the two RasInfos )
    lscreen +scViewport +vpRasInfo ( together. )
    @ +riNext !

    lscreen +scViewport +vpModes dup w@ ( Set Modes for DUALPF )
    DUALPF | swap w!

    lscreen MakeScreen ( This calls MakeVPort() for us. )
    RethinkDisplay ( Calls MrgCop() and LoadView(). )

    lscreen : ( Return the screen addr to caller. )

```



```

\ -----
\ freedual:
\ Synopsis: Restores a dual playfield to single playfield
\           status, and frees memory for the second bitmap.
\ Inputs:   1) screen: Address of an opened intuition
\           screen.
\           2) dualstruct: Address of a dualscrn struct.
\ Returns:  Nothing.

: freedual ( screen / dualstruct )
  locals| dual lscreen |

( Take out the second RasInfo )
  0 lscreen +scViewPort +vpRasInfo @ +riNext !

( Turn off the DUALPF bit )
  lscreen +scViewPort +vpModes dup w@
  DUALPF xor swap w!

( Re-do the display )
  lscreen MakeScreen
  Rethinkdisplay

( Free the display memory. )
  dual +dsWidth w@
  dual +dsHeight w@
  dual +dsDepth w@
  dual +dsBitMap2 freebitmap ;

\ *****
\ ** Miscellaneous variables and functions used by *
\ ** the gel support routines.
\ *****

\ ***** The array gelhandles holds ALL dynamically allocated
\ ***** memoryhandles. That's why it has to be so large.

100 4 larray gelhandles
global handleindex 0 to handleindex

\ ***** These arrays and structures are used by the
\ ***** system in the Gelsinfo structure.

create nextline 16 allot nextline 16 erase
create lastcolor 32 allot lastcolor 32 erase

struct CollTable &collisions
struct GelsInfo ginfo
struct VirtualSprite head
struct VirtualSprite tail

\ *****
\ ** readygels:
\ ** 1) Sets NO sprites reserved.
\ ** 2) Sets pointers to nextline and lastcolor.
\ ** 3) Sets pointer to CollTable.
\ ** 4) Sets gel boundary collision coordinates.
\ ** 5) Links gelsinfo to rastport.
\ ** 6) Calls InitGels.

: readygels ( gelsinfo / rastport - )
  locals| lrast lginfo |

  -1 lginfo +gisprRsrvd c!
  nextline lginfo +giNextLine !
  lastcolor lginfo +giLastColor !

  &collisions lginfo +giCollHandler !

  0 lginfo +giLeftMost w!
  5 lginfo +giTopMost w!
  dwidth lginfo +giRightMost w!
  dheight lginfo +giBottomMost w!

  lginfo lrast +rpGelsInfo !
  head tail lginfo InitGels
  WaitTOF ;

\ *****
\ ** These flags tell whether or not the user wants
\ ** double buffering, and what type of gel to allocate.

1 constant DBUFF
0 constant SPRITEF
1 constant VSPRITEF
2 constant BOBF
3 constant ANIMCOMPFF
4 constant ANIMOBFF
5 constant BORDERLINEF
6 constant SHADOWF
7 constant SAVE&DBUFF
8 constant DBUFFPACKF
9 constant IMAGEF
10 constant SPRCOLORSF

: allocate ( width height depth type spriteflag - handle )
  locals| sprtflag type ldepth lheight lwidth |

```

```

type
case SPRITEF of SimpleSprite CLEAR get.memory
endof
VSPRITEF of VirtualSprite CLEAR get.memory
endof
BOBF of Bob CLEAR get.memory
endof
ANIMCOMPFF of AnimComp CLEAR get.memory
endof
ANIMOBFF of AnimOb CLEAR get.memory
endof
BORDERLINEF of lwidth 2* CHIP CLEAR | get.memory
endof
SHADOWF of lwidth lheight * 2* CHIP CLEAR | get.memory
endof
SAVE&DBUFF of lwidth 16* lheight ldepth * AllocRaster
endof
DBUFFPACKF of DBufPacket CLEAR get.memory
endof
IMAGEF of lwidth lheight * ldepth * 2*
  sprtflag
  if 8+
  then
  CHIP CLEAR | get.memory
  endof
SPRCOLORSF of 6 CLEAR get.memory
endof
endcase

?dup
if
  SAVE&DBUFF type = not
  if dup handleindex gelhandles !
    handleindex 1+ to handleindex
  then
  else
    ." Memory allocation error!"
    ." Gel type requested was: #" type . cr
    ." handle index is: " handleindex . cr abort
  then ;

\ *****
\ ** Routines for reading in of gel data. One is
\ ** for VSprites, Sprites and Bobs, the other is
\ ** for AnimObs only.

: readgelfile ( filename / vsprite / spriteflag - )
  0 0 locals| imagehandle lfileid sprtflag lvsprt filename
  |

  filename MODE_OLDFILE dosOpen dup
  if
    to lfileid
    lfileid lvsprt VirtualSprite dosRead drop
    lvsprt +vsWidth w@
    lvsprt +vsHeight w@
    lvsprt +vsDepth w@
    IMAGEF sprtflag allocate to imagehandle

    lfileid imagehandle @ imagehandle handle.size
    sprtflag
    if
      8- swap 4+ swap
    then
    dosRead drop
    imagehandle @ lvsprt +vsImageData !
  else
    drop
    ." i/o error in readgelfile. Check filenames." cr
    ?turnkey if bye
    else abort
    then
  then
  lfileid dosClose ;

: readacbf ( lvsprt lanimcomp lanimob filename - )
  0 0 locals| imagehandle lfileid filename lanimob lanimcomp lvsprt |

  filename MODE_OLDFILE dosOpen dup
  if
    to lfileid
    lfileid lvsprt VirtualSprite dosRead drop

    lvsprt +vsWidth w@
    lvsprt +vsHeight w@
    lvsprt +vsDepth w@
    IMAGEF 0 allocate to imagehandle
    lfileid imagehandle @ imagehandle handle.size dosRead drop

    imagehandle @ lvsprt +vsImageData !

    lfileid lanimcomp AnimComp dosRead drop
    lanimob if
      lfileid lanimob AnimOb dosRead drop
    then
    lfileid dosClose
  else

```

(continued)



```

drop
." i/o error in readaobfile. Check filenames" cr
?turnkey if bye
else abort
then
then ;

\ *****
\ ** !sequence is used by Sprites, VSprites and bobs
\ ** to store a pointer to alternate imagedata in an
\ ** array specified by the user.
\ *****

: !sequence ( filename / addr / flag - )
0 0 locals| lvsprt handle sprtflag seqarr filename |

VirtualSprite CLEAR get.memory dup to handle
@ to lvsprt

filename lvsprt sprtflag readgelfile
lvsprt +vsImageData @ seqarr !
handle to.heap ;

\ *****
\ ** freegels: Loops through gelhandles array, freeing
\ ** space, if an element is non-zero.
\ *****

: freegels ( - )
handleindex 0 do
    i gelhandles @ dup 0=
    if drop
    else to.heap
    then
    loop
0 to handleindex ;

```

### Listing Three: dblstruct.f

```

\ Listing Three: dblstruct.f
\ Routines for creating a double buffered Intuition display...
\
\ dbufscrn is a consolidation of variables associated with
\ double buffered displays. Note that this replaces the structure
\ called 'bmappack' in 'screens.misc'. If you're using double
\ buffering, you don't need 'bmappack'.

structure dbufscrn
    SHORT: +dbsbmWidth ( width of the bitmaps )
    SHORT: +dbsbmHeight ( their height )
    SHORT: +dbsbmDepth ( depth )
    BitMap STRUCT: +dbsBmap0 ( The structures themselves. )
    BitMap STRUCT: +dbsBmap1 ( Note, NOT pointers )
structure.end

\
\ 'frametog' is used by 'togglemaps' for switching bitmaps...
\ The user need never bother with it.

global frametog 0 to frametog

\ togglemaps:
\ Synopsis: Switches between two bitmaps, according to the
\ following scheme: if 'frametog' is a 1, draw
\ into BitMap 1, display BitMap 0.
\ else
\ drawinto BitMap 0, display BitMap 1.
\
\ Inputs: 1) rport: Address of a RastPort struct, gotten from
\ an opened screen.
\ 2) rinfo: Address of a RasInfo, from the viewport
\ of an opened screen.
\ 3) dbufscrn: Address of a dbufscrn structure
\
\ Returns: Nothing.

: togglemaps ( rport / rinfo / dbufscrn struct )
locals| dbuf rinfo rport |

frametog
if
    ( store Bmap 0 to be displayed )
    dbuf +dbsBmap0 rinfo +riBitMap !
    ( store Bmap 1 to be drawn into )
    dbuf +dbsBmap1 rport +rpBitMap !
else ( do the opposite of above )
    dbuf +dbsBmap0 rport +rpBitMap !
    dbuf +dbsBmap1 rinfo +riBitMap !
then

currentscreen @ MakeScreen
RethinkDisplay
frametog not to frametog ;

```

```

\
\ makedbuf:
\ Synopsis: Create a double buffered intuition display.
\ Call this in place of 'onscreen'.
\
\ Inputs: 1) NewScreen: Address of a NewScreen struct.
\ 2) dbuf : Address of a dbufscrn struct.
\ 3) cmapfile : Null terminated name of a color map file.
\ Send a 0 here to indicate no color file.
\ 4) #cregs : Number of color registers in the cmap.
\ 5) ctable : Address of the array holding color values.
\
\ Returns: Address of opened screen.

: makedbuf
( newscreen / dbuf / cmapfile / #cregs / ctable - scrn add )
locals| ctable l#cregs cmapfile dbuf lnscreen |

dbuf +dbsbmWidth w@
dbuf +dbsbmHeight w@
dbuf +dbsbmDepth w@
dbuf +dbsBmap0 makebitmap
lnscreen +nsCustomBitMap !

lnscreen OpenScreen

cmapfile if
    cmapfile l#cregs ctable readcmap
    currentscreen @ +scviewport swap l#cregs
    LoadRGB4
    then

dbuf +dbsbmWidth w@ ( make the second BitMap )
dbuf +dbsbmHeight w@
dbuf +dbsbmDepth w@
dbuf +dbsBmap1 makebitmap drop

( The following starts us off drawing into BitMap 0. )
currentscreen @ dup +scRastPort swap
+scViewport +vpRasInfo @ dbuf
togglemaps

currentscreen @ ;

\
\ offdbuf:
\ Synopsis: Closes and frees up display memory for a double
\ buffered Intuition display.
\ Inputs: 1) screen: address of a double buffered screen.
\ 2) dbuf : address of a dbufscrn structure.
\ Returns: Nothing
\
\ This is to be called in place of offscreen, in 'screens.misc'.

: offdbuf ( screen dbufscrn - )
locals| dbuf lscreen |

lscreen CloseScreen

dbuf +dbsbmWidth w@
dbuf +dbsbmHeight w@
dbuf +dbsbmDepth w@
dbuf +dbsBmap0 freebitmap

dbuf +dbsbmWidth w@
dbuf +dbsbmHeight w@
dbuf +dbsbmDepth w@
dbuf +dbsBmap1 freebitmap ;

```

### Listing Four: dualstruct.f

```

\ -> Listing Four: singlepf.f
\ -> Example of putting up a simple Intuition screen.

include df1:misc4th/screens.misc

\ ----- Constants and Globals -----
320 constant dispwidth \ Width of the display
200 constant dispheight \ Height
4 constant dispdepth \ depth
16 constant #cregs \ Number of Color registers in cmap.

create cmapfile 0, "mf-files:spiro.cmap" \ Name of the cmap file
create colortable #cregs 2* allot \ The array to hold color
\ Values.

global rastport ( Global variable for easy access )
global viewport ( ditto )
global rasinfo ( ditto. But not really needed if )
( you're not doing any scrolling. )

\ ----- Structures -----
\ Create and initialize a bmappack, and newscreen struct.
\ These will be sent to the routines in 'screens.misc'.

```

(continued on page 99)



# AmiExpo Midwest '88

## Chicago Hosts a Very Happy Third Birthday for the Amiga!

by Michael T. Cabral

AmiExpo Midwest '88 proved that the entire country has now been "Amigatized." Naysayers snickered that the first AmiExpo in New York would fall on its face because the Amiga just wasn't big enough. AmiExpo Los Angeles was haunted by predictions of tiny crowds and less than bustling booths. Questions about mid-country interest and general ho-hum ushered in AmiExpo Chicago. But, if the AmiExpo shows are any indication, the Amiga express has arrived at the top floor of the computing world.

AmiExpo New York was a huge inaugural success. AmiExpo Los Angeles drew flesh-on-flesh crowds and rabid interest. With 10,429 attendees over the three days, AmiExpo Chicago spread the frosting on the Amiga's third birthday cake...both literally and figuratively.

The literal spreading took place at a gala celebration of the Amiga's third birthday after day two of the show. Black and red balloons blanketed the Regency Ballroom and acted as acoustic trampolines for a rousing "Happy Birthday To You" dedicated to everyone's favorite inanimate bundle of chips. Under the twinkle of flash bulbs, Jay Miner carved the first slice of the mountainous cake and the celebration blasted off.

More than a third year of survival was being celebrated, though. What was really being celebrated was what had gone on down the hall for the previous eight hours—the enthusiastic display of tons of hardware and software. The Amiga may have made it to the ripe old age of three, but we all know that without hardware and software support any machine is doomed to die a slow and painful death. All the new hardware developments and software solutions introduced at AmiExpo Midwest formed the sweetest possible frosting for the Amiga's third birthday cake and guaranteed more celebrations to come.

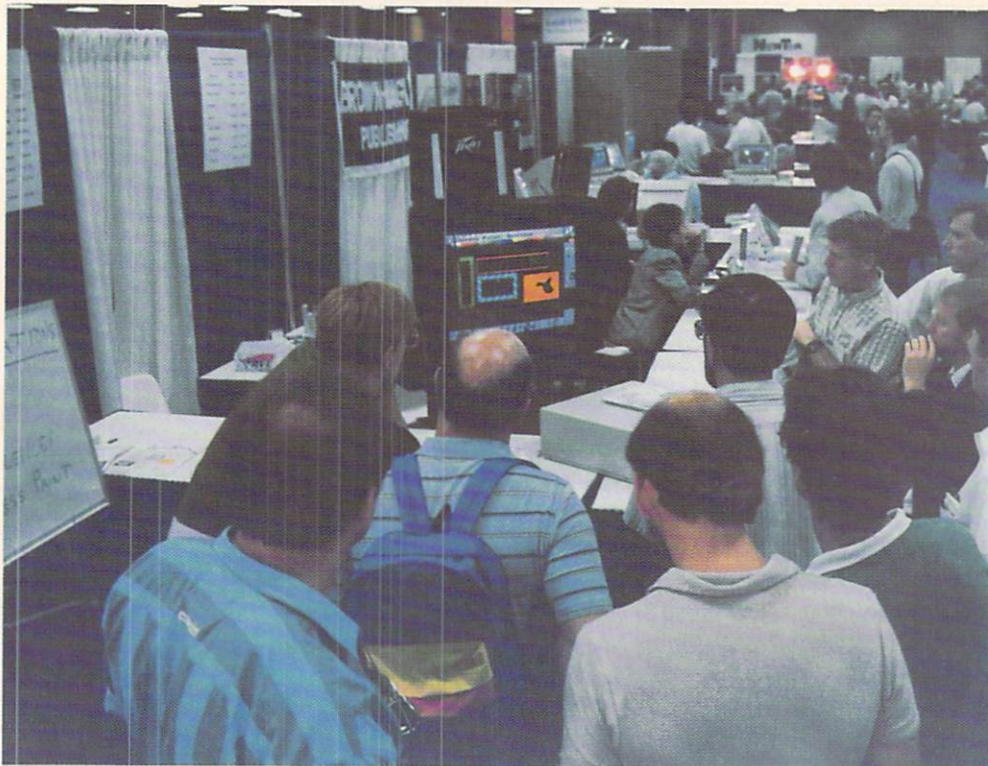
### Commodore's Contribution

As much as the Amiga needs hardware and software, hardware and software need the Amiga. One hand washes the other, right? So the big question at the two previous AmiExpos has been, "When is Commodore going to get their %&\*%! hands in the sink?" New York and Los Angeles were each graced only by a few Commodore techies and luminaries roaming the floor. No booth, no display, no real support.

(continued)







*Brown-Wagh created some bustle, demonstrating products like excellence! and Express Paint.*

In Chicago, the previously invisible appeared with a bang. When the familiar "Only Amiga" promo theme song blasted the doors open for day one of AmiExpo Chicago, no one in the hall could deny that Commodore was there. The centrally-located booth was predictably the show's largest and part of a new, nationwide reach-out-and-touch campaign. According to a rep assigned to the Midwest on a full-time basis, Commodore has scattered public relations people throughout the country for full-time, grass roots work. Dealers will be covered with more than the usual token visits and the Amiga bug will be put in the ears of all those who aren't already infested. The effort is targeted at making Amiga a household name.

Fresh seemed to be the order of the show for Commodore as they also sported a brand new twenty-minute video. The video is longer and more informational than the "Test Flight" clip, with much less flash and glitz. The narrator seated on the edge of a desk talking about desktop video (what else?) and other serious topics offered a sure sign that Commodore has finally decided

to market the Amiga as much more than a game arcade for your living room.

What else was in the booth? If you're a pessimist, the booth was sort of empty... or if you're an optimist, there was plenty of room for talking, mingling, and getting to know the Amiga. Either way, it was just nice to see Commodore. The reps stationed at the booth full-time for questions, handshaking, and some good ole-fashioned PR were more than enough to make the booth a welcome, popular attraction.

The highlight of the display was a beta of the A2024 multisync monitor. The hi-res display looked sharp and very Mac-like.

The beta was also shown alongside the Byte by Byte booth by Amiga original Dale Luck, but despite the double exposure, a tentative release date was out of the question.

Along the left side of the Commodore booth, an A500 ran a video showoff implementing A-Squared's Live! digitizer, Mimetics' AmiGen genlock, and Elan Design's Invision real-time image manipulation software. Titling was shown with Zuma's TV\*Text running on another A500 with a 30MB SupraDrive, and Gold Disk's Professional Page ran a tidy layout on an NEC multisync monitor. Output capabilities, always prominently in question, were answered this time with glossy color prints from Tektronix 4693D and 4896 printers. CSA hardware products were also on display uninstalled, further cementing the cozy spot they've formed in all of the Commodore show booths.

Enough about Commodore. On to the guys who put the Amiga to work.



*Dr. Oxide "operates" on an unsuspecting visitor.*



### Busy Booths

From the beginning, one of the Amiga's bragging points has been its Intuition interface. Simple, easy-to-use, and lets you work like you think. **Emerald Intelligence** is ready to make the Amiga truly intuitive with Magellan, an artificial intelligence/expert system building tool introduced at AmiExpo.

Magellan promises to turn your Amiga into "a powerful knowledge engineering workstation, offering performance and interface features unavailable on the IBM-PC, PS/2 or Apple Macintosh." Wow! The capabilities for really putting the Amiga to work here are mind-bending. Without learning the ins and outs of C, or even BASIC, you can combine your practical knowledge with the power of your Amiga to exploit desktop video, networking, hi-res graphics, and more. Whatever your Amiga can do, Magellan can help you get at it.

Magellan lets you think and work within the framework of the IF-THEN, on-the-go logic you implement in everyday situations. Like Macintosh HyperCard, Magellan lets you build according to the raw blueprint your brain maps out. The introductory version of Magellan hits the market for \$195.

With Perry Kivolowitz and the **ASDG** contingent demonstrating a lot of something for everybody, their booth rarely stopped for a breath. Top billing was split between the eagerly awaited Twin-X board and the JX-450 Controller for the SpectraScan scanner.

Due to market with a \$329 price tag right about the time you read this, the Twin-X impressively ran a 110 volt AC switch, a bar code reader, and a IEEE 488 (GPIB) module in the booth. Three modules for the Twin-X were also officially announced. A dual serial module, the SBX-Serial/2, will provide two channel full-duplex asynchronous communications with RS 232 drivers on an IEEE 959. Look for this one around October for a list of \$199. The SBX-SCSI, a complete SCSI controller also on an IEEE 959, should also hit in October for the same \$199. The SBX-GPIB, running

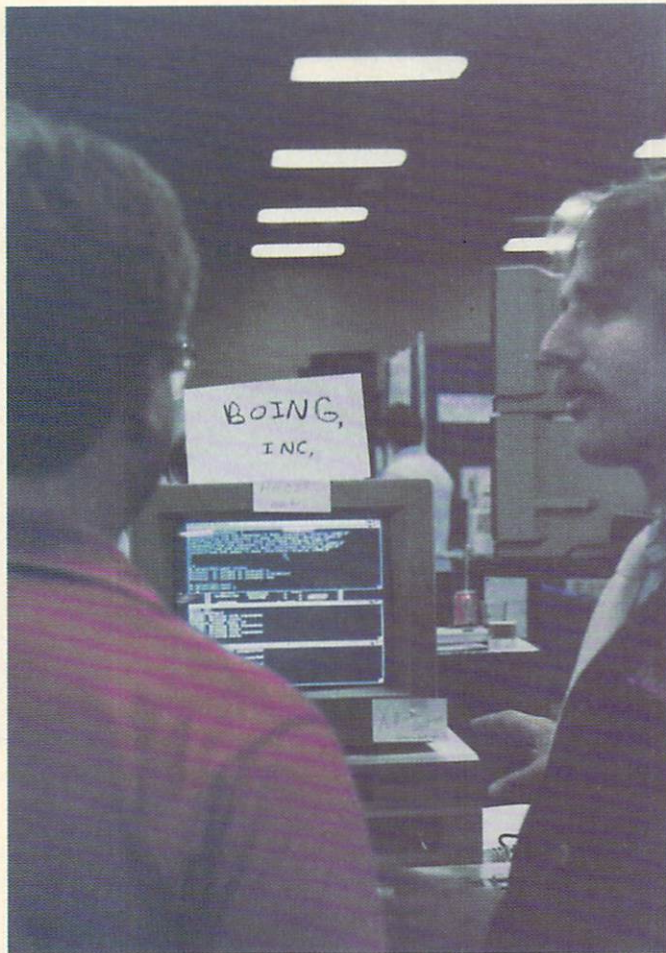
in preliminary form at the show, will, as expected, handle all IEEE 488 1975/1978 functions and the IEEE 488/A supplement. We should see this generic module for \$199 soon after Twin-X hits.

The surprise smash was ASDG's software support for the SpectraScan scanner. The JX-450 Controller, expected to be out in about two months, will allow you to control, alter, and display many scanning options on the Amiga. Output to an H-P Paint Jet printer revealed how nice a team the scanner and software will make. The software will be available solo or with the scanner. If you don't already own a SpectraScan, start pocketing pennies—the package with scanner will run near \$7000!

The ASDG party didn't end there, either. On the software side, CygnusEd Professional, a West German import, made its American debut. Deemed "the new king of editors" in Germany, this editor allows you to edit ten files simultaneously with lightning-fast "turbo" scrolling. CygnusEd includes many of the text editing givens, plus macros, a layout mode for charts, screen resolution of up to 1000 x 800, and ARexx compatibility. Best of all, the program can recover crashed edits in progress!

### Video and Graphics Madness!

As usual, the folks behind all that dazzling Amiga video drew hordes. The bottom line is that you can have a great product in any market, but hot video will pull the curious out of the aisles and into



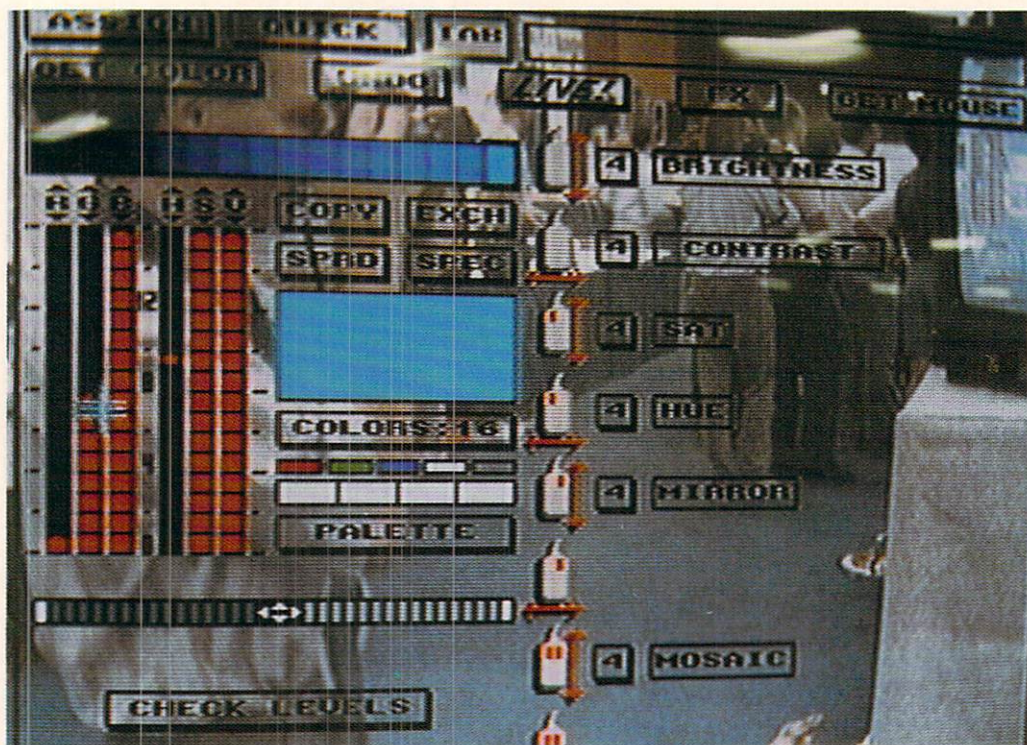
Dale Luck alongside a beta of Commodore's upcoming A2024 monitor.

booths. The **NewTek** booth, just steps away from the main entrance to the hall, gathered its usual flock. Of most interest was the much rumored and oft-misunderstood Video Toaster. The pre-Toaster was demonstrated and some viewers could have used periscopes to get at least a peek. There was even a kitchen toaster on hand toasting away and feeding the busy NewTekkers. NewTek is still promising that the Video Toaster will be "the ultimate desktop video system." The big question: WHEN?!

**Byte by Byte**, although showing nothing new of their own, magnetized crowds of gazers with their usual bouncing, bounding, spinning demos from Sculpt and Animate-3D. The most riveting newcomer was a flawless free-throw shooter netting twine-tickler after twine-tickler. A new package called Fancy 3D Fonts by **Access Technologies** was on sale at the booth since the fonts work with Animate and Sculpt.

(continued)





*Elan Design's Invision dropped graphics over live video of Expo attendees.*

Also nearby because of their ties to Byte by Byte was **Syndesis**. Along with their ingeniously necessary Interchange program, the folks from Syndesis also showed InterFont, a 3D object font designer that was due in late August. InterFont creates 3D fonts for titling in programs like Sculpt-3D, VideoScape 3D, Forms in Flight, and Turbo Silver.

The **Elan Design** booth elicited many double takes. When passersby figured out they were on camera with a George Michael video gyrating over their on-screen image, many stopped to check out the source of the action. The source was Invision, an original real-time special effects system that allows you to interact with video pumped in from A-Squared's Live! digitizer or a VCR. You can mix images from paint programs with the live stuff and incorporate effects like strobe, mosaic, and mirror. Invision is on the shelves now for \$129, but keep in mind that you'll also need the Live! digitizer if you want to toy with live footage.

In addition to appearing in both the Commodore and Elan Design booths, Live! also had a nook of its own. **A-Squared** had Live! for the A500 and

A1000 on hand, along with a promising preview of Live! 2000. How promising? Well with two cameras, Live! 2000 will allow 3-D, real-time video! Along with other wildness, Live! 2000 will pop video into an oval which you can then float across the screen by dragging the mouse. Better yet, the trail you leave behind will be made up of the image within the oval!

The rapidly growing Amiga genlock club also picked up two new high-end members in Chicago. **Magni Systems, Inc.** displayed the 4000 Series of Video Graphics Systems and wowwed many casual strollers. The basic 4000 system boasts true broadcast standard video straight from Amiga graphics. Two plug-in cards combine encoder and genlock to provide controlled edges and timing, correct color framing, software selectable bandwidths, sync and black burst generators, and more. Fades and keying can also be tapped through software.

Interestingly enough, all the way across the arena in a seemingly separate world, **Compu-Art** launched North American distribution of a similar turnkey video workstation manufactured in Australia. The Neriki Image Master Pro-Genlock

(joined with the Fairlight CVI real-time digital effects generator) is also a full broadcast quality genlock ready to tackle the professional video market. Who will win the high end battle? It was a mite difficult to tell at the show—both booths ran the same demo of fading football scores and showed no difference to the naked eye.

The **RGB Computer and Video Creations** booth featured the most elaborate video set-up of the show. Amazingly enough, all the equipment stacked high along the back of the booth was under the control of an A500 and two A2000s. In this system called the Super-VHS Desktop Video Workstation, the A500 acted as a dedicated character generator, while the A2000s managed two video source machines, a special effects

generator, genlocked graphics, and sound. RGB also offered members of their now-legendary Deluxe Help family of interactive tutorials: Deluxe Help for DigiPaint, Deluxe Paint II, Calligrapher, and PhotonPaint. Currently in the works are Deluxe Help modules for PageSetter, SuperBase Professional, City Desk 2.0, Aegis DrawPlus, and yes, AmigaDOS 1.3.

### **Games Gallery**

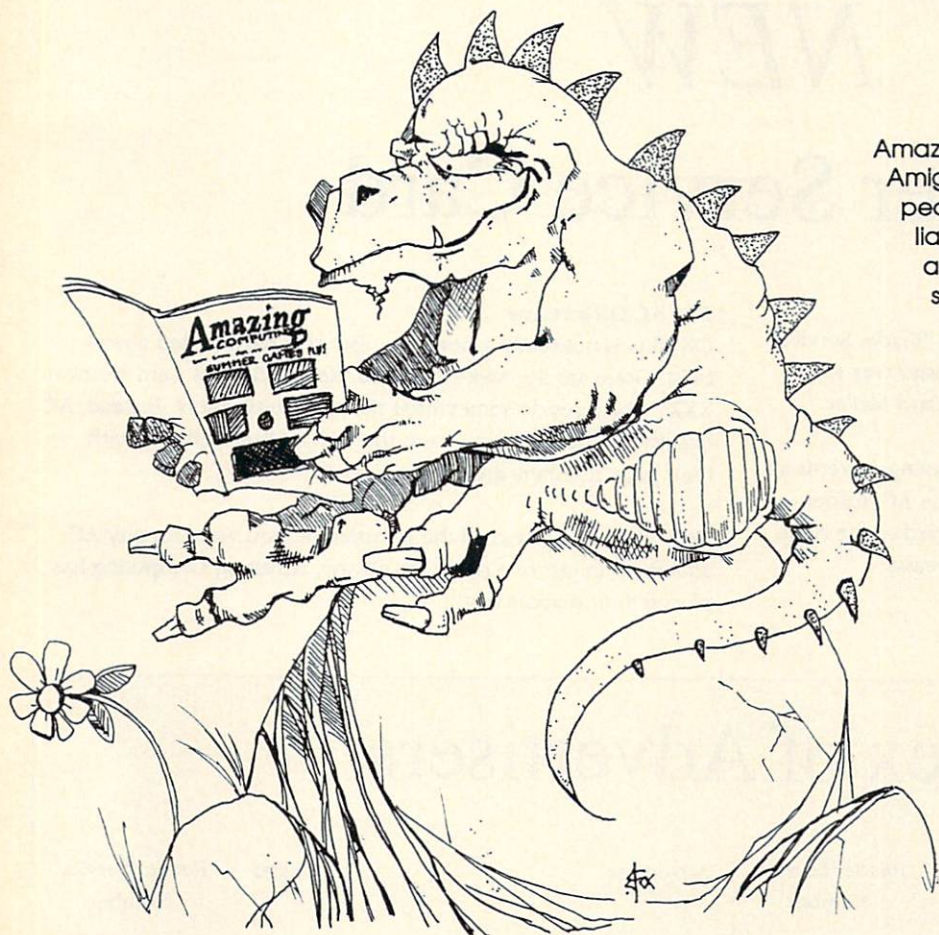
Only one type of booth attracts more wide-eyed attention than a video booth: a games booth. The most action- and people-packed gaming site at AmiExpo was undoubtedly the **Actionware** booth. Along with double light guns blasting away at Capone, the booth also featured the gun-em'-down pace of new releases P.O.W. and Creature. To add flavor to P.O.W., one Actionware rep sported camouflage duds for two days of the show. Talk about realism!

**ReadySoft, Inc.** captured the bulk of the gaming wows on the final day of the show with a pre-release of the animated arcade smash, Dragon's Lair. ReadySoft had the preview pony-expressed in on

*(continued on page 97)*



Shouldn't EVERYONE see your product in the Amazing Product Guide: Winter '89?  
Developers! Last Chance for a FREE listing in our Amazing Product Guide: Winter '89!



Amazing Computing is read far and wide by Amiga enthusiasts. Programmers, business people, and gaming gurus read AC. Australians, Americans, Canadians and Europeans read AC. Even rare, but astute stegosaurus dragons read AC!

What does this spectrum-covering audience mean to you? Simply put, every one of these users should know about your product. That's why we are putting the finishing touches on the *Amazing Product Guide: Winter '89*, an issue devoted solely to all Amiga products. Amiga users will pick up our *Amazing Product Guide: Winter '89* searching for products to put the Amiga to work—your products.

So, if you have not submitted product information for the *Amazing Product Guide: Winter '89*, THIS IS IT!

Your products WILL BE LEFT OUT if you do not respond by:

**September 30, 1988**

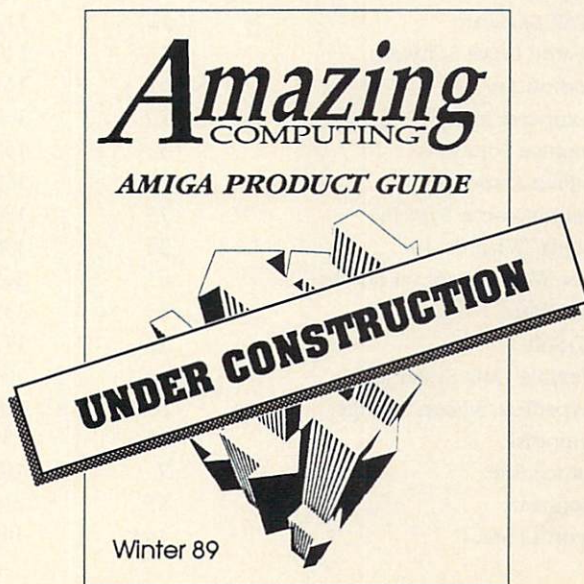
If we already have your information, no problem. But if you suspect that we do not have information about your products, please check with us immediately. Even if you fall into the "no problem" category, please make sure we have your most up-to-date information. Please contact:

Michael Cabral  
*Amazing Product Guide: Winter '89*  
(508) 678-4200

The *Amazing Product Guide: Winter '89*, like all issues of AC, is also a cost-effective way of advertising your product. An audience of ravenous Amiga enthusiasts will be looking for your product in the Guide. They will also be looking for your overall presence in the market. Reinforce your listing in the *Amazing Product Guide: Winter '89* with your best, most informative advertisement.

For advertising information about the *Amazing Product Guide: Winter '89* or any issue of AC, contact:

John Fastino  
Advertising Manager  
(508) 678-4200



Remember, **September 30** is the *Amazing Product Guide: Winter '89* deadline. Don't be left out!



# Amazing Computing's

## NEW

## Reader Service Card

### Wow, AC has done it again!

Due to the tremendous response to our original Reader Service Program, Amazing Computing announces an easier way to contact AC advertisers: The AC Reader Service Card Mailer.

To use the AC system, simply locate all the Amazing Advertisers you wish to contact below, find their appropriate AC Reader Service Card Numbers, and mark them on the card to the right. Fill in your address and mail the card. It is that easy!

### The AC Difference

OK, this is not really a new idea. But, we have added a new twist. Gone are the awkward lines "Reader Service Card Number XXX" which reside somewhere on an advertisement. Instead, AC has listed all the advertisers in the Index Of Advertisers with their page numbers and special codes.

Now it is easy to find all the information you want for any AC advertiser in just one place. As always, Amazing Computing has placed their readers first!

## Index of Advertisers

Advertiser	Page	Reader Service Number	Advertiser	Page	Reader Service Number
A-Squared Distributions, Inc.	1	101	Megatronics	CII	292
Aelen Electronics	48	148	Metropolitan Computer Products	54	154
Amazing Computer Systems, Inc.	43	143	Michigan Software	13	113
AmiExpo	9	109	Micro Way	67	167
BEST Software	12	112	Microbotics, Inc.	50	150
Central Coast Software	11	111	Microsmiths, Inc.	25	125
Compu Art	61	161	New Wave Software	22	122
Computer Mart	70	170	NewTek	CIV	294
Creative Solutions	52	152	OTG Software	52	252
D-Five Associates	63	163	Peacock Systems, Inc.	18	118
Delphi Noetic Systems	73	173	Pioneer Computing	34	134
Digital Wizards, Inc.	38	138	Radical Eye Software	53	153
East West Computer Shows	63	263	Sedona Software	101	201
EaseWare	33	133	Software Advantage Consulting Corp.	104	204
EZ-Soft	77	177	Software Ingenuity	44	144
Flexible Data Systems, Inc.	64	164	Spencer Organization	62	162
HyperTek/Silicon Springs	69	169	The Memory Location	42	142
Jumpdisk	55	155	The Other Guys	5	105
Lattice, Inc.	7	107	Tru-Image	62	162
Lionheart	55	255	Videomaker	19	119
Lynn's Luna C	103	103	William Hawes	59	159



*(AmiExpo, continued from page 94)*

the final day just to get some exposure. One look and you see why they wanted this baby to get air play. When you first see the graphics and animation, you'll take a look around to make sure you didn't somehow stumble into a video arcade. The six disks (that's right, six disks) should retail for about \$50 when the game is released this fall. ReadySoft also announced Cosmic Bouncer, Rock Challenge, and Scary Mutant Space Aliens From Mars as other fall goodies.

This year's award for smartest show of hustle in the gaming market has got to go to **DigiTek Software**. They pump out quality games faster than Stephen King can pump out million-selling spine tinglers. DigiTek announced no less than *nine* pre-September releases to complement established winners like Vampire's Empire and Amegas. The most addictive was Hole-in-One Miniature Golf, a slick combination of fun, skill and the luck of pool table-like bounces. Dropped jaws greeted demo scenes of North Sea Inferno. Also shown were Joe Blade, Western Games, Extensor, Spin-World, SkyBlaster, Cyber Complex, Final Mission, and Qix lookalike, PowerStyx.

Sure games draw crowds, but **Discovery Software International** opted against taking any chances. What did they do? They held a puppet show, of course! High above the booth, the hip Pac-a-like, Zoomer, bopped along to his own little promotional rap for Discovery's Zoom! Appropriately less cute was Cyrus the Virus, a representative of precisely what Discovery's new preventative medicine against viruses, V.I.P., can help avoid. On the all-important subject of viruses, **Abacus Software** also plans to release *Computer Viruses—A High Tech Disease*, a book explaining what a virus is, how it works, and how to protect yourself.

#### **Happy Birthday Dear Amiga, Happy Hardware To You!**

Despite the insulting selling tactics of Dr. Oxide and his disheveled disorderlies, the rubber chicken, rubber rat infested booth known as **Comp-U-Save** did have one redeeming quality: the Bus Expander, unpretentiously manufactured

by **Bill's Boards**. The Bus Expander lets you configure your A500 or A1000 to accept most expansion cards designed for the A2000. Hard disk controllers, RAM cards, and thanks to the A2028 Bridgeboard, even IBM-compatible add-ons are possible. The Expander has six A2000 bus slots with three bridgeable to AT slots and six IBM slots (four AT slots).

**M.A.S.T** stands for Memory and Storage Technology, makers of the Amiga-compatible UniDrive and TwinDrive. A look at the new hardware from these guys shows that their name isn't the only thing they've made smaller. Minimegs 500 is the latest entry in M.A.S.T's line of "matchbox peripherals." No kidding—we're talking tiny hardware. Minimegs can squeeze up to 2MB of auto configuring RAM into a very small box. The Tiny Tiger is a similarly small 45MB SCSI interface and drive. Minimegs goes for \$199 unpopulated, and the 45MB Tiny Tiger with interface, drive case, and power supply runs \$795.

**Spirit Technology Corporation**, an old favorite in the Amiga hardware world, brought along three yet-to-be-released items. The new MIDI Star is calling itself

"the most versatile and complete MIDI interface available." With two INs, six OUTs, an RS-232 port with passthrough, and two-color LED indicators for all ports, Spirit may be right. We'll see when it hits, since the show model was without a case and barely developed enough to run a demo. Spirit also premiered the HDA-506 Hard Drive Adapter that will allow you to use IBM-compatible hard drives and ST-506 controllers on your A500 or A1000. Finally, Spirit announced the S 500-2 internal memory expansion. The board is expandable to 2MB of autoconfiguring fast memory and requires only four chips per half MB.

**Great Valley Products, Inc.**, manufacturers of the IMPACT series of hardware add-ons, announced the IMPACT A500-HD/RAM Controller and the IMPACT A2000-HardCard in 20 and 45 MB versions. The A500-HD/RAM combines a 20MB hard drive and memory expansion subsystem that snaps onto your A500. The unit includes a SCSI controller, the hard drive, and 2MB of autoconfiguring FAST RAM. The SCSI allows you to connect up to seven external devices and includes a Mac

*(continued)*



*Starvision International introduced themselves to the Amiga audience with the Omega Stereo Sampler.*



compatible connector. The internally installed A2000-HardCard20 or 45 affords a SCSI controller and a 20 or 45 MB hard drive without hogging a peripheral slot.

Dabbling in the little bit of everything side of the hardware market is **Creative Microsystems, Inc.** In their booth, Creative offered the Processor Accelerator, the VI-Series of video adapters, and MIDI 1. How's that for diversity? For \$199.95, the accelerator boasts speed better than an MC68010 and near the speed of 68020 boards. A math co-processor slot promises even better speed. The video adaptors convert your Amiga's RGB signal into composite, chroma/luma, or RF signals for use with VCRs, other monitors or TV. For \$79.95, MIDI 1 includes one IN, three selectable OUT/THRU's, an RS-232, SYNC OUT for drum machines, and blinking LEDs.

### **Welcome New Developers!**

For the sound starved, a new company called **Starvision International** demonstrated the Omega Stereo Sampler blasting out more sound than anyone could handle. With a huge poster of a space shuttle as their background, Starvision continually blew everybody's ears off with the actual rumbling, crackling sound of take-off. The cacophony turned disturbed heads all the way across the hall, and developers at nearby booths really got annoyed. Starvision amicably toned things down and thought about using headphones to display the power of the Omega.

From the sound arena, Starvision will move to education with the release of World Atlas (\$49.95) in October, and to games with Mega Pinball (\$39.95) in November, Twin Ranger (\$39.95) and Snowberry (\$29.95) in December, and Starfighter One sometime in 1989.

Music software was scarce in Chicago, so the **New Wave Software** booth drew mobs rivaling both the video and gaming booths and is definitely worth a mention. Even if the hall had been packed with sound stuff, New Wave still would have been hot. Up and running and perking all ears within range were the new Sound Oasis sample disk reader and Dynamic Studio 2.0, complete with sequencer, drum machine, note editor, and added SMPTE support.

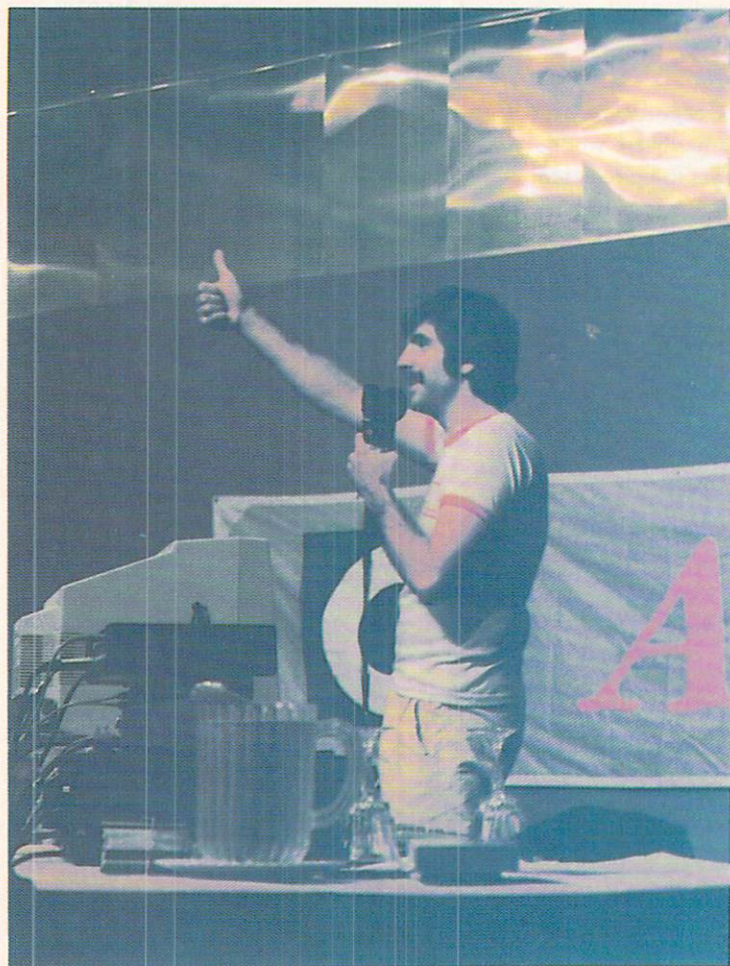
Back on the subject of new companies, **Precision Incorporated** was making final plans to meet the American market on August 15. Precision Software Ltd., the leading manufacturer of data management software in Europe, has launched this new American wing under the guidance of Dan Browning, formerly of **Progressive Peripherals and Software**. The split between Browning and Progressive was a friendly one, and Superbase Personal, Superbase Professional, and Logistix will now be handled directly and exclusively by Precision Inc.

Precision Software Ltd., fresh off a successful conquest of companies like Blythe, Quartz, and Borland at the PC User's Conference Database Challenge, will hit the American market with sales, distribution, and tech support through Precision Inc. Precision Inc. also promises to be a two-way street for North American developers trying to reach the European market. As an additional note, PR man Erich Stein has also amicably left the Progressive nest to go out on his own as an Amiga PR specialist.

As all the AmiExpos have done, the Midwest show spelled out the flourishing state of Amiga hardware and software support. New aspects of video are being developed more avidly than ever. Games, graphics, page layout, sound, hardware design, and virtually everything else the Amiga can do, continue to beat past standards.

Totally new ground is being covered by far-reaching developments like Magellan and the Twin-X. Strong new companies are popping up everywhere. And Commodore's full-fledged appearance completed the picture that's been missing something at the past two AmiExpos. With Commodore and hustling hardware and software developers now hitting on all cylinders, we can expect even happier, healthier (dare I say more profitable) birthdays to come for the Amiga.

•AC•



*R.J. Mical recalls the birth of the Amiga and muses about the future.*



\ screens.misc for the definition of bmapack.

```
struct bmapack bmap
  bmap bmapack erase
  600 bmap +bpWidth w!
  400 bmap +bpHeight w!
  4 bmap +bpDepth w!
structend

struct NewScreen singlepf
  0 singlepf +nsLeftEdge w!
  0 singlepf +nsTopEdge w!
dispwidth singlepf +nsWidth w!
dispheight singlepf +nsHeight w!
dispdepth singlepf +nsDepth w!
  0 singlepf +nsDetailPen c!
  1 singlepf +nsBlockPen c!
SPRITES singlepf +nsViewModes w!
CUSTOMSCREEN SCREENQUIET | CUSTOMBITMAP |
  singlepf +nsType w!
  0 singlepf +nsFont !
  0 singlepf +nsDefaultTitle !
  0 singlepf +nsGadgets !
  0 singlepf +nsCustomBitMap !
structend

\ ----- Example of how this is used. -----
\ : test
\   singlepf bmap cmapfile #cregs colortable onscreen ;
\ -----
```

### Listing Five: dualpf.f

\ Listing Five: dualpf.f  
\ Example of using the Dual Playfield functions...

```
include dfl:misc4th/screens.misc
include dfl:misc4th/dualstruct.f

\ ----- Constants & Globals -----
320 constant dispwidth ( Display width )
200 constant dispheight ( Display height )
2 constant dispdepth ( Display depth )
16 constant #cregs ( Number of color registers in the )
      ( colortable. )

create cmapfile 0,"mf-files:spiro.cmap" \ Name of the color file.
create colortable #cregs 2* allot \ Array to hold colors.

global &rastport ( Globals for easy access to these structures. )
global &viewport
global &rasinfo ( Keep this in, if you want to scroll play- )
      ( field 1. You don't really need it otherwise. )

\ ----- Structures -----
\ Create and initialize structures needed by the functions in
\ 'screens.misc'.

struct bmapack bmap ( For Playfield 1. )
  bmap bmapack erase
  dispwidth bmap +bpWidth w!
  dispheight bmap +bpHeight w!
  2 bmap +bpDepth w!
structend

struct dualscrn dual ( For Playfield 2. )
  dual dualscrn erase
  dispwidth dual +dsWidth w!
  dispheight dual +dsHeight w!
  2 dual +dsDepth w!
structend

struct NewScreen dualpf
  0 dualpf +nsLeftEdge w!
  0 dualpf +nsTopEdge w!
dispwidth dualpf +nsWidth w!
dispheight dualpf +nsHeight w!
dispdepth dualpf +nsDepth w!
  0 dualpf +nsDetailPen c!
  1 dualpf +nsBlockPen c!
SPRITES dualpf +nsViewModes w!
CUSTOMSCREEN SCREENQUIET | CUSTOMBITMAP |
  dualpf +nsType w!
  0 dualpf +nsFont !
  0 dualpf +nsDefaultTitle !
  0 dualpf +nsGadgets !
  0 dualpf +nsCustomBitMap !
structend

\ ----- Example of how it's used -----
\ : dualtest
\   dualpf bmap cmapfile #cregs colortable onscreen
\   dual makeitdual ;
\ -----
```

### Listing Six: dblpf.f

\ Listing Six: dblpf.f  
\ Example of putting up a double-buffered intuition display.

```
include dfl:misc4th/screens.misc
include dfl:misc4th/dblstruct.f

\ ----- Constants and Globals -----
320 constant dispwidth ( display width )
200 constant dispheight ( display height )
4 constant dispdepth ( display depth )
16 constant #cregs ( Number of color registers in the cmap )

create cmapfile 0,"mf-files:spiro.cmap" \ Name of the cmap file
create colortable #cregs 2* allot \ Array for color values

global &rastport ( Globals for easy access to these structures. )
global &viewport
global &rasinfo ( Now you need this to pass to the routine )
      ( 'togglemaps', whether or not you're scrolling. )

\ ----- Structures -----
\ Create and initialize these structures which are used by the
\ routines in 'dblstruct.f'

struct dbufscrn maps ( Define dimensions of )
  maps dbufscrn erase ( both BitMaps )
  dispwidth maps +dbsbmWidth w!
  dispheight maps +dbsbmHeight w!
  3 maps +dbsbmDepth w!
structend

struct NewScreen dblpf
  0 dblpf +nsLeftEdge w!
  0 dblpf +nsTopEdge w!
dispwidth dblpf +nsWidth w!
dispheight dblpf +nsHeight w!
dispdepth dblpf +nsDepth w!
  0 dblpf +nsDetailPen c!
  1 dblpf +nsBlockPen c!
SPRITES dblpf +nsViewModes w!
CUSTOMSCREEN SCREENQUIET | CUSTOMBITMAP |
  dblpf +nsType w!
  0 dblpf +nsFont !
  0 dblpf +nsDefaultTitle !
  0 dblpf +nsGadgets !
  0 dblpf +nsCustomBitMap !
structend

\ ----- Examples of how it's used -----
\ : test
\   dblpf maps cmapfile #cregs colortable makedbuf
\   dup dup +scViewport to &viewport
\   +scRastPort to &rastport
\   +scViewport +vpRasInfo @ to &rasinfo ;

\ -----
\ This one moves a white rectangle accross the screen.
\ : doit 0 0 locals| x oldx |
\   \
\   \ &rastport 0 SetRast
\   \ &rastport &rasinfo maps togglemaps
\   \ &rastport 0 SetRast
\   \
\   \ 250 0 do
\   \   &rastport 1 setapen
\   \   &rastport x 50 50 x + 100 rectfill
\   \   &rastport &rasinfo maps togglemaps
\   \   &rastport 0 setapen
\   \   &rastport oldx 50 50 oldx + 100 rectfill
\   \   x to oldx
\   \   x 2+ to x
\   \ 2 +loop ;
```

### Listing Seven: kbdstick.f

\ Listing Seven: kbdstick.f  
\ -> Routines to install a tiny input handler in front of  
\ -> Intuition, to be used to catch raw key events.  
\ -> Also, routines to set up controller port 1 for use  
\ -> with a joystick. You must have included the assembler  
\ -> prior to this file. (what a waste!)

```
include dfl:devices/input.f
include dfl:devices/gameport.f

\ -----
\ struct devicepack is a consolidation of variables for
\ communicating with devices...

structure devicepack
  IOStdRequest STRUCT: +dpIOReq \ For communicating with the
      \ device.
  Interrupt STRUCT: +dpInterrupt \ Holds code pointer for our
      \ little handler.
```

(continued)



```

GamePortTrigger
STRUCT: +dpTrigger \ To set what will trigger a
                \ gameport event.
PTR: +dpEvent \ Our handler moves an address
                \ of an InputEvent here.
BYTE: +dpByteData \ A handy place to store 1 byte
                \ data for some device commands.
structure.end

```

```

\ ----- Functions for creating ports, -----
\ ----- and performing device commands. -----
\ setupdevice:
\ Synopsis: Creates and stores a MsgPort in the IOStdReq,
\           Opens the requested device. Aborts on error.
\ Inputs:  1) lname: Null terminated name of the device.
\           2) lpack: A devicepack struct.
\           3) unit : Unit number of the device.
\ Returns: OpenDevice returns a 0 if successful.

```

```

: setupdevice ( dev name / devicepack / unit - MsgPort addr. )
locals| unit lpack lname |
    0 0 CreatePort dup
    if
        lpack +dpIOReq +ioMessage +mnReplyPort !
        lname unit lpack +dpIOReq 0 OpenDevice
    else
        drop ." CreatePort error." cr
        ?turnkey if bye else abort then
    then ;

```

```

\ -----
\ docommand:
\ Synopsis: Stores necessary data to perform a device command
\           into the IOStdReq structure.
\ Inputs : 1) lcommand: Appropriate device command, gotten from
\           the device's file.
\           2) llength : Length in bytes of data to be moved by
\           the device.
\           3) ldata : Where the device is to find (or store)
\           data.
\           4) IOReq : Address of a IOStdReq.
\ Returns : DoIO returns a 0 if successful, guru meditation on
\ failure... :-{

```

```

: docommand ( command / length / data / IOReq - IOReq)
locals| lreq ldata llength lcommand |
    lcommand lreq +ioCommand w!
    llength lreq +ioLength !
    ldata lreq +ioData !
    lreq DoIo ;

```

```

\ ----- Setting up a little "event handler" -----
\ kbdpack: Globally declared structure, so user won't have to
\ bother to declare his own.
struct devicepack kbdpack
kbdpack devicepack erase

```

```

\ -----
\ setupkbd:
\ Synopsis: Opens input.device, and installs the adds the
\ handler to the chain, in front of Intuition.
\ Inputs:  1) lpack: Address of a devicepack struct.
\           2) code : Address of handler code.
\ Returns: 0 on success, guru on failure.

```

```

: setupkbd locals| code lpack |
    0" input.device" lpack 0 setupdevice
    51 lpack +dpinterrupt +isnode +lnpri c!
    lpack +dpevent
    lpack +dpinterrupt +isdata !
    code lpack +dpinterrupt +iscode !
    ind_addhandler interrupt lpack +dpinterrupt lpack +dpioReq
    docommand ;

```

```

\ -----
\ takedownkbd:
\ Synopsis: Removes the handler from the chain, closes the
\ input.device, and deletes the ReplyPort.
\ Inputs : lpack : Address of the devicepack
\           {kbdpack, most likely}.
\ Returns : None.

```

```

: takedownkbd ( devicepack - )
locals| lpack |
    ind_remhandler interrupt lpack +dpinterrupt lpack +dpIOReq
    docommand
    lpack +dpIOReq CloseDevice
    lpack +dpIOReq +ioMessage +mnReplyPort @ DeletePort ;

```

```

\ -----
\ Handler:
\ Synopsis: Stores the inputevent into the contents of reg. A1.
\           Passes the event unchanged to Intuition.
\ Inputs:  1) A0 : the InputEvent
\           2) A1 : my data area.
\ Returns: D0 : unchanged InputEvent.

```

```

CODE Handler
    A0 A1 () LONG MOVE,
    A0 D0 LONG MOVE,
    RTS,
END-CODE

```

```

\ create Handler
\ 2288 w, 2084 w, 4e75 w,

```

```

\ -----
\ monitorkbd:
\ Synopsis: If there was an input event, and if it was a
\ RAWKEY, we want it. Process all of the events
\ by examining the Next field of the InputEvent
\ struct.
\ Inputs : 1) lpack : Address of a devicepack.
\           2) levent : Local variable, initialized to the
\ address of the InputEvent.
\ Returns: Whatever you want it to.

```

```

: monitorkbd 0 locals| levent lpack |
    lpack +dpevent @ dup
    if
        to levent
        begin
            levent
            while
                levent +ieClass c@ ieClass_rawkey =
                if
                    currentscreen @ dup
                    if DisplayBeep
                        else drop
                        then
                            ( Your code here. )
                            ( Probably something like storing )
                            ( which key was pressed. )
                        then
                            levent @ to levent
                        repeat
                    else drop
                    then ;

```

```

\ ----- End of Keyboard -----
\ ----- Start on Joystick -----
\ Globally declared structures, so user won't have to worry
\ about declaring them himself.

```

```

struct devicepack stickpack
struct InputEvent stickevent
stickevent InputEvent erase \ Extremely important to clear
stickpack devicepack erase \ memory!

```

```

\ -----
\ setupstick:
\ Synopsis: Opens gameport.device on unit 1, sets the port
\ to an abs_joystick, and trigger events to come
\ only when the stick changes position.
\ Inputs:  lpack: Address of a devicepack.
\ Returns: 0 on success.

```

```

: setupstick ( devpack - )
locals| lpack |
    0" gameport.device" lpack 1 setupdevice
    not if
        gpct_absjoystick lpack +dpByteData c!
        gpd_setctype 1 lpack +dpByteData lpack +dpIOReq docommand
        not if
            gptf_downkeys gptf_upkeys +
                lpack +dpTrigger +gptKeys w!
                1 lpack +dpTrigger +gptTimeout w!
                1 lpack +dpTrigger +gptXDelta w!
                1 lpack +dpTrigger +gptYDelta w!
            gpd_settrigger GamePortTrigger lpack +dpTrigger
            lpack +dpIOReq docommand
        else
            ." Error setting up stick." cr
            ?turnkey if bye else abort then
        then
    else
        ." Error setting up device" cr
        ?turnkey if bye else abort then
    then ;

```

```

\ -----
\ Globally declared X, Y, and Trigger variables.
global dx \ change in X.

```



```

global dy \ change in Y.
global ?trig \ Was the button pressed?
\
\ monitorstick:
\ Synopsis: Reads an event from the gameport.
\ Inputs : 1) lpack : Address of a devicepack for the
\           gameport.
\           2) levent: Address of an InputEvent struct to
\           store data in.
\ Returns : Data from event is in the InputEvent struct you
\           passed.
: monitorstick ( devpack / stickevent - )
    locals| levent lpack |

    gpd_readevent InputEvent levent lpack +dpIOReq
    docommand drop

    levent +ieX w@ to dx
    levent +ieY w@ to dy

    levent +ieCode w@ iecode_lbutton -
    if
        1 to ?trig
    then
        levent +ieCode w@ iecode_lbutton iecode_up_prefix + -
        if
            0 to ?trig
        then ;

\
\ takedownstick:
\ Synopsis: Closes gameport.device, Calls DeletePort,
\           and sets controller type to GPCT_NOCONTROLLER.
\ Inputs : lpack : Address of the devicepack struct.
\ Returns : Nothing.
: takedownstick ( lpack - )
    locals| lpack |
    GPCT_NOCONTROLLER lpack +dpByteData c!
    GPD_SETCTYPE 1 lpack +dpByteData lpack +dpIOReq docommand
    drop

    lpack +dpIOReq CloseDevice
    lpack +dpIOReq +ioMessage +mnReplyPort @ DeletePort ;

```

### Listing Eight: prog.f

```

\ -> Listing eight: prog.f
\ -> Example of opening a screen, monitoring the keyboard
\ -> and joystick, and of scrolling the display.

include dfl:misc4th/singlepf.f
include dfl:misc4th/kbdstick.f

: setupscreen ( open a single playfield screen )
    singlepf bmap cmapfile #cregs colortable onscreen
    dup dup
    +scRastPort to &rastport ( store addresses in global )
    +scViewPort to &viewport ( variables for easy access. )
    +scViewPort +vpRasinfo @ to &rasinfo ;

: initdevices ( open devices, install kbd handler )
    kbdpack ' handler 4- setupkbd
    not if
        stickpack setupstick drop
    then ;

: scrollx ( increment or clip scrolling in the X direction )
    dx 1 >
    if
        -1 to dx
    then
        &rasinfo +riRXOffset w@ dx 2 * + dup
        0< swap bmap +bpwidth w@ dispwidth - > or
        if
            &rasinfo +riRXOffset w@
        else
            &rasinfo +riRXOffset w@ dx 2 * +
        then ;

: scrolly ( increment or clip scrolling in the Y direction )
    dy 1 >
    if
        -1 to dy
    then
        &rasinfo +riRYOffset w@ dy 2 * + dup
        0< swap bmap +bpheight w@ dispheight - 10 - > or
        if
            &rasinfo +riRYOffset w@
        else
            &rasinfo +riRYOffset w@ dy 2 * +
        then ;

: test

```

## Money Mentor™

"Keep track of your pennies, and your dollars will take care of themselves." Old, but sensible advice, even in today's financial environment. Money Mentor™ is a breakthrough in personal financial management. It harnesses the power of the Amiga™ to compute and graph clear reports of your financial situation.

A unique system called "Smart Scrolls" handles diverse data entry functions and can save 70% of the typing typically required for entry.

### Money Mentor™ features:

- 200 budget categories.
- 30 integrated accounts: checking, cash, saving and credit cards.
- Elaborate search routine allows editing of transactions according to your specific guidelines.
- Automatic check printing & account balancing.
- Colorful graphic reports illustrating actual versus budgeted amounts.
- Over 50 reports from which to choose.

This year... get organized with MoneyMentor™

## SEDONA SOFTWARE

11844 Rancho Benardo Rd., Ste. #20  
San Diego, CA 92128

To order, call (619) 451-0151

```

setupscreen
initdevices

```

```

0 &rasinfo +riRXOffset w!
0 &rasinfo +riRYOffset w!

```

```

&rastport 1 SetAPen
&rastport 1 1 100 100 RectFill

```

```

&rastport 2 SetAPen
&rastport 0 0 Move
&rastport 590 350 Draw

```

```

&rastport 6 SetAPen
&rastport 400 250 450 300 RectFill

```

```

begin
    kbdpack monitorkbd ( process kbd events )
    stickpack stickevent monitorstick ( get stick movements )

    dx if ( check for scroll in X direction )
        scrolly
        &rasinfo +riRXOffset w!
    then
        dy if ( check for scroll in Y direction )
            scrolly
            &rasinfo +riRYOffset w!
        then
            dx dy or if ( we only need to do these )
                WaitTOF ( if stick movement has occurred. )
                &viewport ScrollVPort
                currentscreen @ MakeScreen
                ReThinkDisplay
            then
                ?trig ( loop until trigger pressed )
            until

    currentscreen @ bmap offscreen ( take down all our deeds )
    kbdpack takedownkbd 2drop
    stickpack takedownstick

    ?turnkey if bye else abort then ;

```

•AC•



# R O O M E R S

by The Bandito

Atari is heading for trouble. Most of its revenues come from the lucrative videogame market, where Atari claims to hold a 30% share in the U.S. (Nintendo has most of the rest with Sega a distant third.) But as the Bandito reported earlier, videogame mania in Japan is tapering off, and news over here is that videogames may have about a year (or less) left in their life-cycle. (The 11-year-olds are getting bored.) This is bad news for Atari because videogames have been propping up their business for the last two years. U.S. ST sales have fizzled, and the Amiga is rapidly eroding Atari's market share in Europe.

Atari's attempts to market a PC clone have been unsuccessful (unlike Commodore's successful entries in the U.S. clone field). Atari's new ST models, the Mega ST's, have flopped, made worse by the rising DRAM prices which prevented Atari from keeping prices as low as they wanted. In fact, Atari was so upset, they sued Micron, one of their chip suppliers (later settled out of court). Atari's purchase of the Federated stores has been a millstone around their necks, dragging profits down to a miniscule level. Atari's talk about transputers and UNIX workstations hasn't exactly got Sun or Apple shivering in their boots.

They'd better come up with something before the videogame bubble bursts, or Uncle Jack could be looking for a new business. The Bandito hears there's an opportunity to make money in the toy business...

So if Atari and the ST are taking on water, heading for a rendezvous with the

*Titanic*, the Amiga should have smooth sailing in the market, right? Wrong. There's an iceberg on the horizon, and it's shaped like a giant Apple. The Bandito can now reveal the details of the collision course, due to some electronic luck.

The Bandito was playing with his modem during the recent solar storm when a surprising connection occurred due to the disruption of the ionosphere. Somehow, the Bandito's little Amiga accessed the mighty Cray of Apple Computer! Of course, as an Amiga partisan would surely note, the Cray doesn't have a blitter chip or stereo sound, but nonetheless it's still a reasonably powerful computer. So the Bandito did some digging around and, among a lot of boring programs about chip design and data files on circuit layouts, found a file called "Amiga-Killer" which made very interesting reading.

What's this, you say? Apple has finally taken notice of the Amiga? Well, of course Apple has followed the Amiga very closely. The Apple IIx was in development for years before the Amiga came out. The Amiga beat Apple's computer to the marketplace, and Apple watched closely to see what happened. They evaluated the Amiga's features and knew it was superior to the IIx, but it was too late to substantially change their own design. So they released their IIx as the IIGS, figuring marketing could make up for the hardware flaws.

When the Apple IIGS first came out, the press ballyhooed that it would kill the Amiga and the ST. Well, the ST may have

committed suicide, but it soon became clear that the IIGS wasn't fit to process the Amiga's keyboard I/O, if you catch my drift. Even the Apple partisans complained that the IIGS was too slow, had no animation support (not even page-flipping), and the much-touted 15 voice sound ("just like an Ensoniq Mirage synthesizer") was buzzy and suffered from a lack of memory. Developers complained about the lack of system software, no tools or C compilers, and almost no help on Apple's part. Finally, the IIGS cost a whopping \$2500 for a two-drive, one megabyte system with an RGB monitor (that's *without* a printer).

So hardware-wise, the machine had nothing going for it. All it had was Apple's marketing organization. And that, dear friends, has been enough to sell about 400,000 IIGS's so far, which compares very favorably (dollarwise) with Commodore's total of 600,000 Amigas. In fact, the Bandito will bet you a one megabit RAM chip that Apple's profit margins are significantly higher than Commodore's.

But that's not the whole story. The IIGS constitutes a better market for software and hardware add-ons, since almost all sales have been in the U.S., while about half of Amiga's sales have been scattered over Europe. True, a sizable chunk of IIGS sales have gone into schools that don't buy much software. But the Software Publisher's Association reports that software sales for IIGS format have been higher than Amiga format. So both software and hardware manufacturers are eagerly supporting the IIGS. (Hardware manufacturers like the slots provided, as



opposed to the Amiga 1000 with no standard card cage.)

So far the fight has been fairly even: Commodore's superior hardware versus Apple's superior marketing. But Apple isn't one to let someone win a chunk of *their* market. They've come out swinging against IBM's dominance in the business market, and now Macintosh is beating IBM's PS/2 line in sales. Now Apple is turning their sights to the Amiga.

Amiga's last chance to establish its market will be this Christmas, because Apple is going to introduce their Amiga-Killer in January. The IIGS Plus will have a 7 MHz 65C816 CPU running at the same clock speed as the Amiga's 68000. So what's the big deal? After all, the 68000 is a 16/32 bit processor, while the 65816 is only an 8/16, right? But remember, the 65816 is based on the 6502 and is more like a RISC processor than anything else. No wait states for the memory in this puppy—it needs memory that runs as fast as it does. Many instructions that take three cycles on the Amiga will take only one on the IIGS Plus. Functionally, it will be like a 68000 running at 14 MHz.

The IIGS Plus also offers new screen resolutions: 320 x 200 with 256 colors out of 4096, 320 x 400 (interlace) with 256, 640 x 200 or 640 x 400 (interlace) x 16 colors. They're offering more colors than the standard Amiga display modes, though not as many as HAM (but without the restrictions of HAM mode). They've also added interlace modes to go after the video market. The IIGS already has a composite video out, and Apple's making it just as close to NTSC as the Amiga's (in fact, they use the same chip). Note that to get composite video out of an Amiga 500 or 2000 you have to spend fifty bucks for an adapter.

They've fixed the sound problems of the IIGS by re-engineering the motherboard (removing the interference caused by poor circuitry design) and getting a new power supply that doesn't put out the interference the old one did. The IIGS has only 64K of memory available for digitized instrument sounds, and that

memory must be used in blocks of powers of two (4K, 8K, 16K, or 32K). Functionally, you could only have two reasonable instruments at a time.

The IIGS Plus will have 128K memory for digitized sound (separate from machine RAM), and moving instruments in and out will be so fast you can have a whole symphony orchestra playing. Full stereo is available with a \$40 add-in card, so you can have fifteen voice sound with true Ensoniq Mirage quality in stereo! Far better than the Amiga's four voice sound.

The IIGS Plus will have a toolbox ROM of 256K, just like the Mac, for an easy programming environment. (They're upgrading the system software and the development tools—developers can work on Mac II's and just move the code over on disk.) For memory, one megabyte system RAM will be the standard when RAM prices drop, but until then, the machine will come with 512K. Finally, the IIGS Plus has a huge software base—all the old Apple II

programs and a growing number of GS specific software, much of which originally appeared on the Amiga.

What are the weaknesses of this wonder machine? Well, it doesn't have a blitter chip, so theoretically it's less powerful in the animation arena. But given the power of the CPU, developers claim you won't notice any difference, at least for animation-intensive game programs. Only 256 colors at once are possible, instead of 4096. Most HAM pictures don't use more than 256 colors anyway. And without the fringing problems of HAM mode, images on the IIGS Plus should look better than on the Amiga. There's no multi-tasking, but as long as memory is so expensive, no one's likely to notice. Once they bring the system software up to the Macintosh level, they'll probably start multi-tasking or similar features by the time RAM gets cheap enough that you can use multi-tasking. By the way, the IIGS Plus will use very fast RAM to match the processor's speed, so its RAM will be faster than the Amiga's.

(continued)



Workbench	RAM Tools	WBExtras
	Open	WBExecute
	Close	ViewText
	Load	SeePicture
	Unload	LoadPicture
		UnloadPicture

**WBEXTRAS**  
by Peter Dunlap

**Isn't it time you  
got the most from  
your Amiga?**

Now, "WBExtras" is here and is specifically designed to enhance operation of the Amiga by the "New User" as well as the "Seasoned Programmer".

**WBExtras Provides . . .**

- \* New Menus for Workbench Access to Tools and Programs in RAM Disk.
- \* 14 "New" Workbench Tools and 13 Program Source Files.
- \* Complete System Control through Workbench Icons and Menus.
- \* Full Inter-Program Communication with "Parameter Passing."
- \* Menu Access to Multiple Icon Selection.
- \* Reduced Multitasking Memory Usage.
- \* User Selectable Audio Response.
- \* Indexed Interactive Multi-Level Tutorial Screens.
- \* Full Documentation, Examples, Customizing Suggestions and Glossary.

**Indeed, WBExtras is an essential for every Amiga User  
and a necessity for anyone with a Hard Disk System!**



P.O. Box 1308  
Cañon City, CO 81212  
719 275-5858

**Amiga & AmigaBasic Workbench TM of Commodore - Amiga, Inc.**

*\*\*Dealer Inquiries Invited\*\**

**\$3995**

Plus \$3.00 for Shipping  
Colorado Res. Add Sales Tax





## Even up the Score !!!



Start using your Amiga to give you the Advantage in making better investment decisions! Color Graphics of Individual Stocks and General Market Trends. High Low Close, Moving Averages, Volume, Momentum and Relative Strength. Select the best performers with Relative Strength Rankings. Know when to get into and out of the market. Update stocks and/or mutual funds manually or automatically. Easy to use Communications included.

Only \$99.95

See your local dealer or call:

Software Advantage Consulting Corporation  
37346 Charter Oaks Blvd  
Mt. Clemens, MI 48043 (313) 463-4995

Amiga and The Investor's Advantage are trademarks of their companies.

Here's Apple's marketing strategy: the Apple IIe will be dropped, and the current IIGS will take its place at about \$1500 for a system. The Apple IIc gets a faster processor and a 3.5" disk drive, maybe even full IIGS compatibility, and an LCD screen option to make it a true laptop. The IIGS Plus takes its place as the flagship of the line at \$2500 (with two drives, one megabyte of RAM, and RGB monitor). Of course, there will be an upgrade path between the IIGS and IIGS Plus—a motherboard swap (for maybe \$500), but at least it's an upgrade path. If these prices seem high, remember that Apple dealers discount them and Apple offers rebates frequently, so look for street prices to be 10 to 20 percent lower. Add in an extensive advertising plan with a full set of TV and print ads. Don't forget Apple's network of 2500 dealers, either. With this plan, Apple intends to pass the one million unit mark for the IIGS/IIGS Plus by Christmas 1989.

Well, it doesn't take a rocket scientist to see that this strategy is aimed squarely at

the A500 and A2000. Amiga will have a price advantage, but the performance race will be a toss-up. So what, you say? People *like* the Amiga because it's neat, and because it's ... well, it's an Amiga.

Face it, Commodore, Amiga fans are Amiga fans because they are technology junkies—they like machines that do neat things. They weren't attracted to the Amiga by its marketing, that's for sure.

And despite what you think, the companies developing Amiga software and hardware are doing it because they like the power and performance of the machine, not because Commodore's so nice to them. (What's that? Did the Bandito hear some developers choking on their cornflakes?) Only a few developers are making good money; most are scraping by, still waiting for Amiga to make "the big break." Many developers are jumping ship to work on Mac and IBM products. What's going to happen when they see a machine as sexy as the Amiga with a very supportive develop-

ment environment and superb marketing that outsells the Amiga? They'll drop the Amiga like a hot lawsuit, and before you can say "Money!", they'll be developing IIGS Plus versions of their products. And the fence-sitters who wonder which computer to buy, and were having trouble distinguishing between an Apple IIGS and an Amiga (their IQ must be no higher than room temperature), will suddenly have a clear choice: Apple.

Now, it's not all gloom and doom. Fortunately for Amiga fans, Apple will position their GS computers as "education" machines only. Oh, if you want to buy one for Junior to use at home to do schoolwork, that's OK. But never, ever, use that dirty word in connection with our computer. You know the word—"entertainment" (sshhhhh!). And while you're at it, don't say "business," either—that's what Macintoshes are for. So Apple will not support entertainment or business for the GS line. That's good news for Commodore. Apple will also have a problem colliding with the Macintosh line, since a IIGS plus will look an awful lot more attractive than a Mac II system for doing color work. They'll have their hands full trying to explain the differences to customers.

What can Commodore do? Get aggressive between now and January when Apple introduces the new machine. Position the 500 as the home entertainment machine of choice, and the 2000 as the business machine for video and graphics. Support developers so they push the machine's capabilities even further, particularly with hot video products. Get new models ready soon with either a faster 68000 or a 68020, more colors, and better sound. And they better hurry, because the clock is ticking...

*[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing™ cannot be held responsible for the reports made in this column. Additionally, Apple Computer was contacted. However, at press time, they had not yet responded. -Ed.]*

•AC•



# The AMICUS & Fred Fish

## Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

<p><b>AMICUS Disk 1</b></p> <p><b>ABasic programs: Graphics</b></p> <p>3DSolids 3d solids modeling prog. w/sample data files</p> <p>Books draws books</p> <p>Cubes draws cubes</p> <p>Durer draws pictures in the style of Durer</p> <p>FScape draws fractal landscapes</p> <p>Hidden 3D drawing program, w/hidden line removal</p> <p>JPad simple paint program</p> <p>Optical draw several optical illusions</p> <p>PaintBox simple paint program</p> <p>Shuttle draws the Shuttle in 3d wireframe</p> <p>SpaceArt graphics demo</p> <p>Speaker speech utility</p> <p>Sphere draws spheres</p> <p>Spiral draws color spirals</p> <p>3D function plots</p> <p>Topography artificial topography</p> <p>Wheels draws arde graphics</p> <p>Xenos draws fractal planet landscapes</p> <p><b>ABasic programs: Tools</b></p> <p>AddressBook simple database program for addresses</p> <p>CardFile simple card file database program</p> <p>Demo multiwindow demo</p> <p>KeyCodes shows keycodes for a key you press</p> <p>Menu run many ABasic programs from a menu</p> <p>MoreColors way to get more colors on the screen at once, using aliasing</p> <p>shapes simple color shape designer Speakt speech and narrator demo</p> <p><b>ABasic programs: Games</b></p> <p>BrickOut classic computer brick wall game also known as 'go'</p> <p>Ochello simple shoot-em-up game</p> <p>Saucer simple talking spelling game</p> <p>Spelling simple talking spelling game</p> <p>ToyBox selectable graphics demo</p> <p><b>ABasic programs: Sound</b></p> <p>Entertainer plays that tune</p> <p>HAL3000 pretends it's a real computer</p> <p>Police simple police siren sound</p> <p>SugarPlum plays 'The Dance of the Sugarplum Fairies'</p> <p><b>C programs:</b></p> <p>ATem simple terminal program, S-E</p> <p>deconv aid to compiling with Lattice C opposite of CONVERT for cross developers</p> <p>Dotty source code to the 'dotty' window demo</p> <p>etchx unix-style filename expansion, partial S-O-D</p> <p>fastexp explains use of fast-floating point math</p> <p>fixdate fixes future dates on all files on a disk, S-E</p> <p>freeDraw simple Workbench drawing prog, S-E</p> <p>GfMem graphic memory usage indicator, S-E</p> <p>Grep searches for a given string in a file with ham shows off the hold-and-modify method of color generation</p> <p>IBM2Amiga fast parallel cable transfers between an IBM and an Amiga</p> <p>Mandel Mandelbrot set program, S-E</p> <p>more patterned graphic demo, S-E</p> <p>objfx makes Lattice C object file symbols visible to Wack, S-E</p> <p>quick quick sort strings routine</p> <p>raw example sample window I/O</p> <p>setface turns on interface mode, S-E</p> <p>sparks qix-type graphic demo, S-E</p> <p><b>Other executable programs:</b></p> <p>SpeechT speech demonstration</p> <p>WhichFont displays all available fonts</p> <p><b>Texts:</b></p> <p>68020 describes 68020 speedup board from CSA</p> <p>Aliases explains uses of the ASSIGN command</p> <p>Bugs known bug list in Lattice C 3.02</p> <p>CLICard reference card for AmigaDOS CLI</p> <p>CLICards guide to using the CLI</p> <p>Commands shorter guide to AmigaDOS CLI commands</p> <p>EdCommands guide to the ED editor</p> <p>FileNames AmigaDOS filename wildcard conventions</p> <p>HalBright explains rare graphics chips that can do more colors</p> <p>ModemPins description of the serial port pinout</p> <p>RAMdisks tips on setting up your RAM: disk</p> <p>ROMWack tips on using ROMWack</p> <p>Wack sound explanation of instrument demo sound file format</p> <p>Speed refutation of Amiga's CPU and custom chip speed</p> <p>WackCms tips on using Wack</p>	<p><b>AMICUS Disk 2</b></p> <p><b>C programs:</b></p> <p>AmigaDOS object library manager, S-E</p> <p>asb test file archive program, S-E</p> <p>auto-chops executable files</p> <p>simple CLI shell, S-E</p> <p>sq, usq file compression programs, S-E</p> <p>Yacht a familiar game, S-E</p> <p>Make a simple 'make' programming utility, S-E</p> <p>Emacs an early version of the Amiga text editor, S-E-D</p> <p><b>Assembler programs:</b></p> <p>bsearch.asm binary search code</p> <p>qsort.asm Unix compatible qsort() function, source and C test program</p> <p>semp.asm sejmp() code for Lattice 3.02</p> <p>SVprint Unix system V compatible printf()</p> <p>tree.c Unix compatible tree() function, O-D</p> <p>(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)</p> <p><b>John Draper Amiga Tutorials:</b></p> <p>Animate describes animation algorithms</p> <p>Gadgets tutorial on gadgets</p> <p>Menus learn about Intuition menus</p> <p><b>AMICUS Disk 3</b></p> <p><b>C programs:</b></p> <p>Xel a C cross-reference gen., S-E</p> <p>Bobol extra-half-bright chip gk demo, S-E</p> <p>Chop truncate (chop) files down to size, S-E</p> <p>CleanUp removes strange characters from text files</p> <p>CRZLF converts carriage returns to line feeds in Amiga files, S-E</p> <p>Error adds compile errors to a C file, S</p> <p>Helo window ex. from the RKM, S</p> <p>KemIt generic KemIt implementation, fakey, no terminal mode, S-E</p> <p>Scalies sound demo plays scales, S-E</p> <p>SkewB Rubik cube demo in hi-res colors, S-E</p> <p><b>AmigaBasicProgs (dir)</b></p> <p>Automata cellular automata simulation</p> <p>CrazyEights card game</p> <p>Graph function graphing programs</p> <p>WitchingHour a game</p> <p><b>ABasic programs:</b></p> <p>Casino games of poker, blackjack, dice, and craps</p> <p>Gomoku also known as 'othello'</p> <p>Sabotage sort of an adventure game</p> <p><b>Executable programs:</b></p> <p>Disassem a 68000 disassembler, E-D</p> <p>DSSide shows a given set of FF pictures, E-D</p> <p>Arrange a text formatting program, E-D</p> <p><b>Assembler programs:</b></p> <p>Argoterm terminal program with speech and Xmodem, S-E</p> <p><b>AMICUS Disk 4</b></p> <p><b>Files from the original Amiga</b></p> <p><b>Technical BBS</b></p> <p>Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.</p> <p>Complete and nearly up-to-date C source to 'image.ed', an early version of the Icon Editor. This is a little flaky, but compiles and runs.</p> <p>An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demomenu3.c, getasac.c, idemo.c, idemo2.c, idemo3.c, idemo4.c, idemo5.c, idemo6.c, idemo7.c, idemo8.c, idemo9.c, idemo10.c, idemo11.c, idemo12.c, idemo13.c, idemo14.c, idemo15.c, idemo16.c, idemo17.c, idemo18.c, idemo19.c, idemo20.c, idemo21.c, idemo22.c, idemo23.c, idemo24.c, idemo25.c, idemo26.c, idemo27.c, idemo28.c, idemo29.c, idemo30.c, idemo31.c, idemo32.c, idemo33.c, idemo34.c, idemo35.c, idemo36.c, idemo37.c, idemo38.c, idemo39.c, idemo40.c, idemo41.c, idemo42.c, idemo43.c, idemo44.c, idemo45.c, idemo46.c, idemo47.c, idemo48.c, idemo49.c, idemo50.c, idemo51.c, idemo52.c, idemo53.c, idemo54.c, idemo55.c, idemo56.c, idemo57.c, idemo58.c, idemo59.c, idemo60.c, idemo61.c, idemo62.c, idemo63.c, idemo64.c, idemo65.c, idemo66.c, idemo67.c, idemo68.c, idemo69.c, idemo70.c, idemo71.c, idemo72.c, idemo73.c, idemo74.c, idemo75.c, idemo76.c, idemo77.c, idemo78.c, idemo79.c, idemo80.c, idemo81.c, idemo82.c, idemo83.c, idemo84.c, idemo85.c, idemo86.c, idemo87.c, idemo88.c, idemo89.c, idemo90.c, idemo91.c, idemo92.c, idemo93.c, idemo94.c, idemo95.c, idemo96.c, idemo97.c, idemo98.c, idemo99.c, idemo100.c</p>	<p><b>Texts:</b></p> <p>serial port commands</p> <p>example of serial port use</p> <p>sample printer interface code</p> <p>printer device definitions</p> <p>region test program</p> <p>source to interface on/off program</p> <p>set the attributes of the parallel port</p> <p>set the attributes (parity, data rate) of the single playfield example</p> <p>source to narrator and phonetics demo</p> <p>simple timer demo</p> <p>exec support timer functions</p> <p>more exec support timer functions</p> <p>loads and displays all available system fonts</p> <p>process() and pbase() assembler include files</p> <p>autogetr.bt warnings of deadlocks with autorequests</p> <p>consoleio.bt copy of the RKM console I/O chapter</p> <p>diskfont.bt warning of disk font loading bug</p> <p>listofnbt list of defines, macros, functions</p> <p>inputdev.bt preliminary copy of the input device chapter</p> <p>License information on Workbench distribution</p> <p>printer pre-release copy of the chapter on printer drivers, from RKM 1.1 v11d.bt diff of fd file changes from version 1.0 to 1.1</p> <p>v2bvt.diff diff of file changes from version 2.0 to 1.0</p> <p><b>AMICUS Disk 5</b></p> <p><b>Files from the Amiga Link / Amiga Information Network</b></p> <p>Note that some of these files are old, and refer to older versions of the operating system. These files are from AmigaLink. For a time, Commodore supported AmigaLink, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.</p> <p>A demo of Intuition menu called 'menuitem', in C source</p> <p>whereis.c find a file searching all subdirectories</p> <p>bobtest.c BOB programming example</p> <p>sound synths example</p> <p><b>Assembler files:</b></p> <p>mydev.asm sample device driver</p> <p>mylib.asm sample library example</p> <p>mylib.i asm-suppl.i</p> <p>asm-suppl.i assembler include files</p> <p><b>Texts:</b></p> <p>amigatalks tips on CLI commands</p> <p>external disk specification</p> <p>gameport port spec</p> <p>parallel port spec</p> <p>serial port spec</p> <p>v1.1.update list of new features in version 1.1</p> <p>v1.1h.bt diff of include file changes to version 1.1</p> <p><b>Files for building your own printer drivers, including dospcalc.c, epsontalk.c, intasm, printer.c, printerlink, printerag.asm, render.c, and wait.asm.</b> This disk contains a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.</p> <p><b>AMICUS Disk 6</b></p> <p><b>IFF Pictures</b></p> <p>This disk includes the DPSide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file at the click of an icon. The pictures include a screen from ArtFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marie Madness, the Bugs Bunny Martini, a still from an old movie, the Dire Straits moving company, a screen from Pinball Construction Set, a TV newscaster, the PaintCan, a world map, a Porsche, a shuttle mission patch, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.</p> <p><b>AMICUS Disk 7</b></p> <p><b>DigiView HAM demo picture disk</b></p> <p>This disk has pictures from the DigiView hold-and-modify video digitizer. It includes the ladies with pencils and lollipops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seelion' program, to turn any screen into an IFF picture.</p> <p><b>AMICUS Disk 8</b></p> <p><b>C programs:</b></p> <p>Browse view text files on a disk, using menus S-E-D</p> <p>Crunch removes comments and white space from C files, S-E</p> <p>EXEQUITE a series of commands from Workbench S-E</p> <p>IconExec EXEQUITE a series of commands from Workbench S-E</p> <p>PDScreen Dump dumps Rasputin of highest screen to printer</p> <p>SetAlternate sets a second image for an icon, when clicked once S-E</p> <p>SetWindow makes windows for a CLI program to run under Workbench S-E</p> <p>SmallClock a small digital clock in a window menu bar</p> <p>Scrimper the screen printer in the fourth AC S-E</p>	<p><b>Amiga Basic Programs:</b></p> <p>(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)</p> <p>AddressBook a simple address book database</p> <p>Ball draws a ball</p> <p>Cloud program to convert CompuServe hex files to binary, S-D</p> <p>Clue a simple clue, intuition driven</p> <p>ColorArt art drawing program</p> <p>DeluxeDraw the drawing program in the 3rd AC, S-D</p> <p>Eliza conversational computer psychologist the game, as known as 'go'</p> <p>Ohello 3D rat maze game</p> <p>RatMaze 3D rat maze game</p> <p>ROR draws 3D pictures of the space shuttle</p> <p>Shuttle simple spelling program</p> <p>Spelling simple spelling program</p> <p>YoYo word zero-gravity yo-yo game, tracks yo-yo to the mouse</p> <p><b>Executable programs:</b></p> <p>3Dcube Modula-2 demo of a rotating cube</p> <p>AtIcon sets a second icon image, displayed when the icon is clicked</p> <p>AmigaSpell a slow but simple spell checker, E-D</p> <p>arc the ARC file compression program must-have for telecom, E-D</p> <p>Bertrand graphics demo</p> <p>disksalvage prog. to rescue trashed disks, E-D</p> <p>KwikCopy a quick but nasty disk copy program: ignores errors, E-D</p> <p>LibDr lists hunk in an object file E-D</p> <p>SaveIBM saves any screen as IFF pic E-D ??</p> <p>ScreenDump shatters screen dump prog. E only</p> <p>StarTerm version 2.0, term program, Xmodem-E-D</p> <p><b>Texts:</b></p> <p>LatticeMain tips on fixing _main.c in Lattice</p> <p>GDskDrive make your own 5 1/4 drive</p> <p>GuruUdd explains the Guru numbers</p> <p>Lat303bugs Lat303 bugs</p> <p>MFForgeRev user's view of the MicroForge HD</p> <p>PinSpool EXECUTE-based print spool prog.</p> <p><b>.BMAP files:</b></p> <p>These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'list', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathtr', 'mathtr2', 'mathtr3', 'mathtr4', 'mathtr5', 'mathtr6', 'mathtr7', 'mathtr8', 'mathtr9', 'mathtr10', 'mathtr11', 'mathtr12', 'mathtr13', 'mathtr14', 'mathtr15', 'mathtr16', 'mathtr17', 'mathtr18', 'mathtr19', 'mathtr20', 'mathtr21', 'mathtr22', 'mathtr23', 'mathtr24', 'mathtr25', 'mathtr26', 'mathtr27', 'mathtr28', 'mathtr29', 'mathtr30', 'mathtr31', 'mathtr32', 'mathtr33', 'mathtr34', 'mathtr35', 'mathtr36', 'mathtr37', 'mathtr38', 'mathtr39', 'mathtr40', 'mathtr41', 'mathtr42', 'mathtr43', 'mathtr44', 'mathtr45', 'mathtr46', 'mathtr47', 'mathtr48', 'mathtr49', 'mathtr50', 'mathtr51', 'mathtr52', 'mathtr53', 'mathtr54', 'mathtr55', 'mathtr56', 'mathtr57', 'mathtr58', 'mathtr59', 'mathtr60', 'mathtr61', 'mathtr62', 'mathtr63', 'mathtr64', 'mathtr65', 'mathtr66', 'mathtr67', 'mathtr68', 'mathtr69', 'mathtr70', 'mathtr71', 'mathtr72', 'mathtr73', 'mathtr74', 'mathtr75', 'mathtr76', 'mathtr77', 'mathtr78', 'mathtr79', 'mathtr80', 'mathtr81', 'mathtr82', 'mathtr83', 'mathtr84', 'mathtr85', 'mathtr86', 'mathtr87', 'mathtr88', 'mathtr89', 'mathtr90', 'mathtr91', 'mathtr92', 'mathtr93', 'mathtr94', 'mathtr95', 'mathtr96', 'mathtr97', 'mathtr98', 'mathtr99', 'mathtr100'</p> <p><b>AMICUS Disk 9</b></p> <p><b>Amiga Basic Programs:</b></p> <p>FlighSim simple flight simulator program</p> <p>HuePalette explains Hue, Saturation, &amp; Intensity</p> <p>Requester ex. of requesters from Amiga Basic</p> <p>ScrollDemo demonstrates scrolling capabilities</p> <p>SynthSizer sound program</p> <p>WorldMap draws a map of the world</p> <p><b>Executable programs:</b></p> <p>Boing! latest Boing! demo, with selectable speed, E</p> <p>Brush2C converts an IFF brush to C data instructions, initialization code, E</p> <p>Brush2Icon converts IFF brush to an icon, E</p> <p>Dazzle graphics demo, tracks to mouse, E</p> <p>DeciGEL assembler program for stopping 68010 errors, S-E-D</p> <p>Klook menu-bar clock and date display, E</p> <p>life the game of life, E</p> <p>TimeSet Intuition-based way to set the time &amp; date</p> <p>EMEmacs another Emacs, more oriented to word processing, S-E-D</p> <p>MyCLI a CLI shell, works without the Workbench, S-E-D</p> <p><b>Texts:</b></p> <p>FractKeys read function keys from Amiga Basic</p> <p>HackerGin explains how to win the game 'hacker'</p> <p>Is68010 guide to installing a 68010 in your Amiga</p> <p>Boing! latest Boing! demo, with selectable speed, E</p> <p>Boing! converts an IFF brush to C data instructions, initialization code, E</p> <p>Brush2Icon converts IFF brush to an icon, E</p> <p>Dazzle graphics demo, tracks to mouse, E</p> <p>DeciGEL assembler program for stopping 68010 errors, S-E-D</p> <p>Klook menu-bar clock and date display, E</p> <p>life the game of life, E</p> <p>TimeSet Intuition-based way to set the time &amp; date</p> <p>EMEmacs another Emacs, more oriented to word processing, S-E-D</p> <p>MyCLI a CLI shell, works without the Workbench, S-E-D</p>
---	--	---	--



<p><b>Tests:</b></p> <p><b>FrncKeys</b> explains how to read function keys from Amiga Basic</p> <p><b>HackerSh</b> explains how to win the game 'hacker'</p> <p><b>Is530</b> guide to installing a 53010 in your Amiga</p> <p><b>PrinterTip</b> sending escape sequences to your printer</p> <p><b>StartupTip</b> tips on setting up your startup-sequence file</p> <p><b>XtmiReview</b> list of Transformer programs that work</p> <p><b>Printer Drivers:</b></p> <p>Printer drivers for the Canon PJ-1080A, the C-10h Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-300, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P100x family, and the Smith-Corona D300, with a document describing the installation process.</p> <p><b>AMC Disk 10</b> Instrument sound demos</p> <p>This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, a kick drum, a marimba, a bass guitar, a flute, a harp arpeggio, a klock drum, a water drip, electric guitar, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibraphone, a violin, a wailing guitar, a horse whinny, and a whistle.</p> <p><b>AMC Disk 11</b> C programs</p> <p><b>drutil</b> Intuition-based, CLI replacement manager</p> <p><b>cpri</b> shows and adjusts priority of CLI processes, S-E</p> <p><b>ps</b> shows info on CLI processes, S-E</p> <p><b>victus</b> displays CompuServe RLE pics, S-E</p> <p><b>AmigaBasic programs</b></p> <p><b>pointed</b> pointer and sprite editor program</p> <p><b>optimize</b> optimization ex. sample from AC article</p> <p><b>calendar</b> large, animated calendar, diary and date book program</p> <p><b>amortize</b> loan amortizations</p> <p><b>brush2BOB</b> converts small IFF brushes to AmigaBasic BOB OBJECTS</p> <p><b>grids</b> draw and play waveforms</p> <p><b>hilbert</b> draws Hilbert curves</p> <p><b>mailib</b> mail lib story generator</p> <p><b>mailtalk</b> talking mailing list program</p> <p><b>meadows3D</b> 3D graphics program, from A C7M article</p> <p><b>mouseback</b> mouse tracking example in hires mode</p> <p><b>slot</b> slot machine game</p> <p><b>toctact</b> the game</p> <p><b>switch</b> pachinko-like game</p> <p><b>weird</b> makes strange sounds</p> <p><b>Executable programs</b></p> <p><b>cp</b> unix-like copy command, E</p> <p><b>cis</b> screen clear, S-E</p> <p><b>diff</b> unix-like stream editor uses 'diff' output to fix files</p> <p><b>pm</b> chart recorder performances indicator</p> <p><b>Assembler programs</b></p> <p><b>cis</b> screen clear and CLI arguments example</p> <p><b>Module-2</b></p> <p><b>trails</b> moving-worm graphics demo</p> <p><b>caseconvert</b> converts Module-2 keywords to uppercase</p> <p><b>Forti</b> Bresenham circle algorithm example</p> <p><b>Analyze</b> 12 templates for the spreadsheet. Analyze</p> <p>There are four programs here that read Commodore 64 picture files. They can translate Koolha Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.</p> <p><b>AMC Disk 12</b> Executable programs</p> <p><b>blink</b> 'blink' compatible linker, but faster, E-D</p> <p><b>clean</b> spins the disk for disk cleaners, E-D</p> <p><b>epconset</b> sends Epson settings to PAR from menu E-D</p> <p><b>showbig</b> view hi-res pics in low-res superbmap, E-D</p> <p><b>teletime</b> tell the time, E-D</p> <p><b>undelete</b> undeletes a file, E-D</p> <p><b>crvaphdm</b> converts Apple II low, medium and high res pictures to IFF, E-D</p> <p><b>menued</b> menu editor produces C code for menus, E-D</p> <p><b>quick</b> quick disk-to-disk nibble copier, E-D</p> <p><b>quickEA</b> copies Electronic Arts disks, removes protection, E-D</p> <p><b>bed 1.3</b> demo of text editor from Microsmiths, E-D</p> <p><b>C programs</b></p> <p><b>spn3</b> rotating blocks graphics demo, S-E-D</p> <p><b>popci</b> start a new CLI at the press of a button, like Sidekick, S-E-D</p> <p><b>vsprite</b> VSprite example code from Commodore, S-E-D</p> <p><b>AmigaBBS</b> Amiga Basic bulletin board prog., S-D</p> <p><b>Assembler programs</b></p> <p><b>star10</b> makes star fields like Star Trek intro, S-E-D</p> <p><b>Pictures</b></p> <p><b>MountMandelbrot</b> 3D view of Mandelbrot set</p> <p><b>Star Destroyer</b> hi-res Star Wars starship</p> <p><b>Robot</b> robot arm grabbing a cylinder</p> <p><b>Tests</b></p> <p><b>vendors</b> Amiga vendors, names, addresses</p> <p><b>cardco</b> fixes to early Cardco memory boards</p> <p><b>cross-reference</b> to C include files</p> <p><b>clues</b> to playing the game well</p> <p><b>make your own slideshows</b> from the Kaleidoscope disk</p> <p><b>AMC Disk 13</b> Amiga Basic programs</p> <p>Routines from Carolyn Schepner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected Convert-D program. With example pictures, and the SaveILBM screen capture program.</p> <p>Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With</p>	<p>documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.</p> <p><b>Executable programs</b></p> <p><b>gravity</b> Sci Amer Jan 85 gravitation graphic simulation, S-E-D</p> <p><b>Tests</b></p> <p><b>MIDI</b> make your own MIDI instrument interface, with documentation and a hi-res schematic picture.</p> <p><b>AMC Disk 14</b> Several programs from Amazing Computing issues:</p> <p><b>Tools</b></p> <p><b>Dan Kany's</b> C structure index program, S-E-D</p> <p><b>Amiga Basic programs:</b></p> <p><b>BMAP Reader</b> by Tim Jones</p> <p><b>FFBrush2BOB</b> by Mike Swinger</p> <p><b>AutoRequester</b> example</p> <p><b>DOSHelper</b> Windows help system for CLI commands, S-E-D</p> <p><b>PETrans</b> translates PET ASCII files to ASCII files, S-E-D</p> <p><b>C Squared</b> Graphics program from Scientific American, Sept 86, S-E-D</p> <p><b>addrs</b> adds or removes carriage returns from files, S-E-D</p> <p><b>dpdecode</b> decrypts Deluxe Paint, memo</p> <p><b>protection</b>, E-D</p> <p><b>asks</b> Yes or No from the user returns exit code, S-E</p> <p><b>vc</b> VisiCalc type spreadsheet, no mouse control, E-D</p> <p><b>view</b> views text files with window and gadget, E-D</p> <p><b>slider</b></p> <p><b>Olmg, Spring, ysBoing, Zaing</b> are sprite-based</p> <p><b>Boing!</b> style demos, S-E-D</p> <p><b>CLIClock, sClock, wClock</b> are window border clocks, S-E-D</p> <p><b>Tests</b></p> <p>An article on long-persistence phosphor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.</p> <p><b>AMC Disk 15</b> The C programs include:</p> <p><b>pr</b> a file printing utility, which can print files in the background, and with line numbers and control character filtering.</p> <p><b>fm</b> displays a chart of the blocks allocated on a disk.</p> <p><b>'Ask'</b> questions an 'execute' file, returns an error code to control the execution in that batch file</p> <p><b>'Star'</b> an enhanced version of AmigaDOS 'status' command.</p> <p><b>'Dissolve'</b> random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.</p> <p><b>PopCLIZ</b> invoke new CLI window at the press of a key.</p> <p><b>The executable programs include:</b></p> <p><b>Form</b> file formatting program through the printer driver to select print styles</p> <p><b>'DiskCat'</b> catalogs disks, maintains, sorts, merges lists of disk files</p> <p><b>'PSound'</b> SunRize Industries' sampled sound editor &amp; recorder</p> <p><b>'Iconmaker'</b> makes icons for most programs</p> <p><b>'Fractals'</b> draws great fractal seascapes and mountain scapes.</p> <p><b>'3D Breakout'</b> 3D glasses, create breakout in a new dimension</p> <p><b>'AmigaMonitor'</b> displays lists of open files, tasks, devices and ports in use.</p> <p><b>'Cosmoids'</b> a version of 'asteroids' for the Amiga.</p> <p><b>'Sizzlers'</b> high resolution graphics demo written in Module 2.</p> <p><b>Tests:</b></p> <p><b>'ans.tst'</b> explains escape sequences the CON: device responds to.</p> <p><b>'FKey'</b> includes template for making paper to sit in the tray at the top of the Amiga keyboard.</p> <p><b>'Spawer'</b> programmer's document from Commodore</p> <p><b>Amiga, describes</b> ways to use the Amiga's multitasking capabilities in your own programs.</p> <p><b>AmigaBasic programs:</b></p> <p><b>'Grice'</b> draw sound waveforms, and hear them played.</p> <p><b>'Light'</b> a version of the Tron light-cycle video game.</p> <p><b>'MigSol'</b> a game of solitaire.</p> <p><b>'Stats'</b> program to calculate betting averages</p> <p><b>'Money'</b> "try to grab all the bags of money that you can."</p> <p><b>AMC Disk 15</b> also includes two beautiful IFF pictures, of the enemy walkers from the box planet in Star Wars, and a picture of a cheetah.</p> <p><b>AMC Disk 16</b></p> <p><b>'Juggler'</b> demo by Eric Graham, a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.</p> <p><b>IFF pictures</b></p> <p>pandies of the covers of Amiga World and Amazing Computing magazines.</p> <p><b>C programs:</b></p> <p><b>'InputHandler'</b> example of making an input handler.</p> <p><b>'FileZip3'</b> binary file editing program</p> <p><b>'ShowPrint'</b> displays IFF picture, and prints it</p> <p><b>'Gen'</b> program indexes and retrieves C structures and variables declared in the Amiga include file system.</p> <p><b>Executable Programs:</b></p> <p><b>'FixHunk2'</b> repairs an executable program file for expanded memory</p> <p><b>'ms2mus'</b> converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.</p> <p><b>'Missile'</b> Amiga version of the 'Missile Command' video game,</p>	<p>This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park</p> <p><b>AMC Disk 17</b> Telecommunications disk which contains six terminal programs.</p> <p><b>'Comm'</b> V1.33 term prog. with Xmodem, WModem, term prog. includes Super Kermit</p> <p><b>'ATerm'</b> V7.2 Dave Wecker's VT-100 emulator with Xmodem, Kermit, and scripting</p> <p><b>'Vt-100'</b> V2.6 VAD(50) port of the Unix C-Kermit</p> <p><b>'Amiga Kermit'</b> Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression</p> <p><b>'Vtek'</b> V2.3.1 V0.3 for CompuServe. Includes RLE graphics abilities &amp; CIS-B file transfer protocol.</p> <p><b>'AmigaHost'</b> expansion memory necessity</p> <p><b>'FishHunk'</b> removes garbage characters from modern received files</p> <p><b>'FluQ'</b> filters text files from other systems to be read by the Amiga E.C.</p> <p><b>'addmem'</b> executable version for use with mem expansion article in AC V2.1</p> <p><b>'arc'</b> file documentation and a basic tutorial on un'arcing' files for making 'arc' files E.C.</p> <p><b>AMC Disk 18</b> Logo</p> <p><b>Logo</b> Amiga version of the popular computer language, with example programs, E-D</p> <p><b>TVText</b> Demo version of the TVText character generator</p> <p><b>PageSetter</b> Freely distributable versions of the updated PagePrint and PageFF programs for the PageSetter desktop publishing package.</p> <p><b>FullWindow</b> Resizes any CLI window using only CLI commands, E-D</p> <p><b>Life3d</b> 3-D version of Conway's LIFE program, E-D</p> <p><b>Delfisk</b> CLI utility to re-assign a new Workbench disk, S-E-D</p> <p><b>Calendar WKS</b> Lotus-compatible worksheet that makes calendars</p> <p><b>SetKey</b> Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D</p> <p><b>VPG</b> Video pattern generator for aligning monitors, E-D</p> <p><b>HP-10C</b> Hewlett-Packard-like calculator, E-D</p> <p><b>SetPrefs</b> Change the Preferences settings on the fly, in C, S-E-D</p> <p><b>StarProbe</b> Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D</p> <p><b>ROT</b> C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D</p> <p><b>Scat</b> Like Ing, windows on screen run away from the mouse, E-D</p> <p><b>DK</b> Decays the CLI window into dust, in Module 2, S-E-D</p> <p><b>DropShadow2</b> Adds layered shadows to Workbench windows, E-D</p> <p><b>AMC Disk 19</b> This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake part T-shirt logo, a sixteen-color hi-res image of Andy Griffin, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.</p> <p><b>Solve</b> Linear equation solver in assembly language, S-E-D</p> <p><b>Gadgets</b> Bryan Catley's AmigaBasicGadgets, Bryan Catley's AmigaBasic household inventory program, S-D</p> <p><b>Waveform</b> Jim Shields' Waveform WofAmigaBasic, S-D</p> <p><b>DiskLib</b> John Kennan's AmigaBasic disk librarian program, S-D</p> <p><b>Subscripts</b> Ivan Smith's AmigaBasic subscript example, S-D</p> <p><b>String, Boolean</b> C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D</p> <p><b>Skinny C</b> Bob Riemersma's example for making small C programs, S-E-D</p> <p><b>COMALh</b> Make C look like COMAL, E-D</p> <p><b>EmacsKey</b> Makes Emacs function key definitions by Greg Douglas, S-D</p> <p><b>Almon 1.1</b> Snoop on system resource use, E-D</p> <p><b>BTE</b> Bernd's Tale character editor, E-D</p> <p><b>Size</b> CLI program shows the size of a given set of files, E-D</p> <p><b>WinSize</b> CLI window utility resizes current window, S-E-D</p> <p><b>AMC Disk 20</b> Compuator, Decoder Steve Michel AmigaBasic tools, S-D</p> <p><b>BobEd</b> BOB and sprite editor written in C, S-E-D</p> <p><b>SpriteMaster!</b> Sprite editor and animator by Brad Kiefer, E-D</p> <p><b>BitLab</b> Bitler chip exploration C program by Tomas Rokicki, S-E-D</p> <p><b>FFpic</b> Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D</p> <p><b>Bankin</b> Complete home banking program, balance your checkbook! E-D</p> <p><b>AMC Disk 21</b> Target</p> <p><b>Target</b> Makes each mouse click sound like a gunshot, S-E-D</p> <p><b>Sand</b> Simple game of sand that follows the mouse pointer, E-D</p>	<p>PropGadget Harriet Maybeck Tolly's proportional gadget example, S-E</p> <p>EHB Checks to see if you have extra-half-bright graphics, S-E-D</p> <p>Piano Simple piano sound program</p> <p>CalScripts Makes cell animation scripts for Aegis Animator, in AmigaBasic</p> <p>This disk has electronic catalogs for AMICUS disks 1 to 20 and Fian disks 1 to 80. They are viewed with the DiskCat program, included here.</p> <p><b>AMC Disk 22</b> Cycles</p> <p><b>Show_Print</b> Light cycle game, E-D</p> <p><b>Views</b> Views and prints IFF pictures, including larger than screen</p> <p><b>PrDrvGen23</b> Latest version of a printer driver generator</p> <p><b>Animations</b> VoiceScope animations of planes and boiling ball</p> <p><b>Garden</b> Makes fractal gardenscapes</p> <p><b>BasicSorts</b> Examples of binary search and insertion sort in AmigaBasic</p> <p><b>AMC Disk 23</b> An AMICUS disk completely devoted to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the first of playing 'Big Sound' on your Amiga</p> <p><b>Instruments</b> a collection of 25 instruments for playing and creating music. The collection ranges from Canon to Marimba</p> <p><b>List INSTR</b> program to list the instruments DMCS will not load as well as list the origins for any instrument</p> <p><b>Music</b> The collection of 14 Classical pieces</p> <p><b>1812Overture</b> The 16 minute classical lecture complete with Canon!</p> <p><b>Three Amiga Music Players:</b></p> <p><b>SMUSPlay</b></p> <p><b>MusicCra2SMUS</b></p> <p><b>MusicStudio2SMUS</b></p> <p><b>AMC Disk 24</b> Sectrama</p> <p>A disk sector editor for any AmigaDOS file-structured device, recover files from a crashed hard disk. By David Joiner of MicroIllusions</p> <p><b>Iconize</b> Reduces the size of IFF images, compression program, Redcor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.</p> <p><b>CodeDemo</b> Module-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example. Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.</p> <p><b>AmiBug</b> Three examples of assembly language code from Bryan Nesbitt:</p> <p><b>BNTools</b> 1. SetLoc, program to switch interface on/off. 2. Why, replace AmigaDOS CLI Why? 3. Loadit, prog to load a file into memory until a reboot. (Only the most esoteric hackers will find Loadit useful.)</p> <p><b>Monolace</b> CLI program - resets Preferences to several colors of monochrome &amp; interface screens. C source is included, works with DisplayPrel, a CLI program which displays the current Preferences settings.</p> <p><b>BoingMachine</b> A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Movie program, which has the ability to play sounds along with the animation. By Ken Offer</p> <p><b>Daisy</b> Example of using the translator and narrator devices to make the Amiga talk. It is written in C</p> <p><b>QuickFix</b> Script-driven animation and slideshow program files through IFF images</p> <p><b>BMon</b> System monitor AmigaBasic program; perform simple manipulations of memory.</p> <p><b>Moose</b> Random background program, a small window opens with a mouse resembling Bulwer's essaying witty phrases user definable.</p> <p><b>DGCS</b> Deluxe Grocery Construction Set, simple Intuition-based prog for assembling and printing a grocery list.</p> <p><b>The Virus Check</b> directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks.</p> <p><b>AMC Disk 25</b> Nemesis</p> <p><b>Nemesis</b> Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics.</p> <p><b>The KickPlay</b> directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADDMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk.</p> <p><b>KeyBird</b> BASIC prog edits keymaps, adjust the Workbench keymaps or create your own.</p>
---	--	--	---







<p><b>Fred Fish Disk 32</b>          Address Extended address book, AmigaBASIC          Calendar Calendar/reminder program, AmigaBASIC          DosPlus1 First volume of CLI oriented developer tools          DosPlus2 2nd volume of CLI oriented developer tools          Executables only          MacView Views MacPaint pics in Amiga low or high res, no sample pictures, by Scott Evernden          Puzzle Simulation of puzzle with moving squares          ShowHAM View HAM pictures from CLI          Solitaire ABASIC games of Canfield and Klondike, from David Addison          Spin3 Graphics demo of spinning cubes, double-buffered example          Sword Sword of Fallen Angel text adventure game written in Amiga Basic          Trails Leaves a trail behind mouse, in Module 2</p> <p><b>Fred Fish Disk 33</b>          3dstars 3d version of the "stars" program below          Bignap Low-level graphics example scrolls bitmap with ScrollVPort          Dou/gels Double-buffered animation example for BOBs and VSpites          DiskMapper Displays sector allocation of floppy disks          MemView View memory in real time, move with joystick          Ong Bouncing balls demo          Spring Ong, with sound effects          ScreenDump Dumps highest screen or window to the printer          Sub Simple database program from a DECUS tape          Stars Star field demo, like Star Trek          TermPlus Terminal program with capture, library, function keys, Xmodem, C-IS-B protocols          Vt100 Version 2.0 of Dave Wecker's VT-100 emulator, with scripts &amp; function</p> <p><b>Fred Fish Disk 34</b>          Airt Support files for Gimpel's 'tm' syntax checker          Blink PD 'link' compatible linker, faster, better          Browser Updated to FF 18 browser, in Marx, with scroll bars, bug fixes          Btree b-tree data structure examples          Calendar Another version of 'btree'          Less Appointment calendar with alarm          NewFonts Set of 28 new Amiga fonts from Bill Fischer          Pr Background print utility, style options, wildcards          Requester Deluxe Print-type file requester, with sample</p> <p><b>Fred Fish Disk 35</b>          ASENDPacket Example of making asynchronous I/O calls to a DOS handler, written by C-A          ConsoleWindow Example of getting the intuition pointer a CON or RAW window, for 1.2, by C-A</p> <p><b>DrUtl</b> Walk the directory tree, do CLI operations from menus  <b>DrUtl2</b> Another variant of DrUtl  <b>FileRequester</b> Lattice C file requester module, with demo driver, from Charlie Heath  <b>MacView</b> Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden  <b>Pop</b> Simple IFF reader program  <b>PopCLI</b> Sidekick-style program invokes a new CLI, with automatic screen blanking  <b>QuickCopy</b> Devenport disk copiers duplicate copy-protected disks  <b>ScrollPI</b> Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield  <b>SendPacket</b> General purpose subroutine to send AmigaDOS packets  <b>SpriteMaker</b> Sprite editor, can save work as C data structure. Shareware by Ray Larson  <b>Tracker</b> Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson  <b>TriClope 3-D</b> Space invasion game, formerly commercial, now public domain. From Geodesic Publications  <b>Tsize</b> Print total size of all files in subdirectories  <b>Unlief</b> C preprocessor to remove given #ifdef sections of a file, leaving the rest alone. By Dave Yost  <b>Vttest</b> VT-100 emulation test program. Requires a Unix system</p> <p><b>Fred Fish Disk 36</b>          Acp Unix-like 'cp' copy program          Clock Updated version of clock on disk 15          Cah Marx 'tah'-like CLI, history, variables, etc          DietKid Diet planning aid organizes recipes, calories          Echo Improved 'echo' command with color, cursor addressing          FixHurk Fixes programs to let them run in external memory          Fm Maps the sectors a file uses on the disk          KioBench Docs, program to make a single disk that works like a Kickstart and Workbench          Lex Computes Fog, Fiesch, and Kincaid readability of text files          TurneVasion David Addison ABASIC 3D maze perspective game          Vc Valsico-like spreadsheet calculator program          Vt100 Version 2.2 of Dave Wecker's telecom program          YaBeing Ongi style game program shows sprite collision details</p> <p><b>Fred Fish Disk 37</b>          This disk is a port of Timothy Budd's Little Smalltalk system, done by Bill Kinnersey at Washington State University</p> <p><b>Fred Fish Disk 38</b>          CSquared Sep 86 SD American, Circle Squared algorithm          FixObj Strips garbage off Xmodem transferred object files          Handler AmigaDOS handler (device) example from C-A</p>	<p>Hp-10c          IFFEncode Saves the screen as an IFF file          IDump Dumps info about an IFF file          Jsh BDS C-like CLI shell          NewStat STATUS-like program, shows priority, processes          Reversi Game of Reversi, version 6.1          Uldecodes Translate binary files to text, Unix-like programs          Vdraw Drawing program, version 1.14          VmIX MIDI synthesizer voice file program          Window Example of creating a DOS window on a custom screen</p> <p><b>Fred Fish Disk 39</b>          'echo', 'touch', 'tst', 'tst' written in assembler          AnsiEcho Displays HAM images from a ray-tracing program, with example pictures          Display Example device driver source, acts like RAM disk          Driver XLIsp 1.7, executable only</p> <p><b>Fred Fish Disk 40</b>          Ahoist Terminal emulator with Xmodem, Kermit and C-IS-B protocols, function keys, scripts, RLE graphics and conference mode          AmigaMonitor Dynamically displays the machine state, such as open files, active tasks, resources, device status, interrupts, libraries, ports, etc          Arc Popular file compression system, the standard for transferring files          AreaCode Program that decodes area codes into state and locality          Blink 'link' replacement linker, version 6.5          Cosmo An 'asteroids' clone          Dg210 Data General D-210 Terminal emulator          DrUtl Windowed DOS interface program, V.1.4          DOSHelper Windowed AmigaDOS CLI help program          PagePrint Prints text files with headers, page breaks, line numbers</p> <p><b>PopCLI</b> Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2, with more          SpriteEd Sprite editor updates two sprites at a time          X-Spell Spelling checker allows edits to files</p> <p><b>Fred Fish Disk 41</b>          AmigaVenture Create your own text adventure programs in AmigaBasic          Version 2.03 of Dillon's C-sh-like shell          Executable only          Doug Macro based C debugging package update to FF #2          DualPlayField example from CBM, update to intuition manual          GetFile Heath's file requester, with source          LatXref Cross reference of Lattice 3.10 header files          Line Drawing demo program          Sines Changes font used in a CLI window          Vt100 Version 2.3 of the VT-100 terminal program</p> <p><b>Fred Fish Disk 42</b>          This disk contains an Amiga version of MicroGNUMacros</p> <p><b>Fred Fish Disk 43</b>          BasicBorg AmigaBASIC program demos page flipping of a 3D cube          Bbm Demo copy of B.E.S.T. Business Management System          BbsList A list of Amiga Bulletin Board Systems          C Compiler frontends for Marx and Lattice C          Copper A hardware copper list disassembler          InstIFF Converts instruments demo sounds to IFF sampled sounds          PopColours Adjust RGB colors of any screen          SpriteClock Simple clock is displayed on a sprite above all screens          ST Emulator Non-serious Atari ST emulator          WBSun Lets Workbench programs be run from the CLI          Wild Two Unix shell style wild card matching routines</p> <p><b>Fred Fish Disk 44</b>          Icons Miscellaneous icons          NewIFF New IFF material from CBM for sampled voice and music files          RayTracePics The famous ray-tracing pictures, from FF#39, now converted to IFF HAM format for 'much' faster viewing          ViewLBM Displays normal and HAM LBM files</p> <p><b>Fred Fish Disk 45</b>          Clue Clue board game          Make Another 'make', with more features          Pictures Miscellaneous pictures          Update Updates older disk with newer files from another disk          Whereals Searches a disk for files of given name</p> <p><b>Fred Fish Disk 46</b>          Asm Shareware 68010 macro assembler, ROM          Kernel Manual compatible          'execute' file program detects presence of modern          Gadget editor from the Programmers Network          Jive Transforms a file from English to Jive          MyLib A binary only copy of Matt's alternate runtime library. Author: Matt Dillon          Subst Berkeley 'ms' and 'rm' macros for 'proff'          ValSpeak Transforms a file from English to Valley Speak</p> <p><b>Fred Fish Disk 47</b>          3D-Arm Simulation of a robotic arm, very good graphics, teaching tool, including C source          Juggler Eric Graham's stunning HAM animation of a robot juggler          VT-100 Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols</p> <p><b>Fred Fish Disk 48</b>          Bru Alpha version of a hard disk file archiver          Comm Version 1.30 of a terminal emulator with phone directories          Cah Version 2.04 of Matt Dillon's Unix 'tah'-like CLI replacement, including Lattice &amp; Marx C source          DiskPerf Disk benchmark program for Unix and Amiga          Du Computes disk storage of a file or directory          MemWatch Program to watch for programs that trash low memory. It attempts to repair the damage, and puts up a requester to inform you of the damage. From the Software Distillery          Profiler A real-time execution profiler for Marx C programs. Includes C source</p>	<p><b>Fred Fish Disk 49</b>          Cycloids Update of electronic spirograph from disk 27          DrUtl Enhanced version of DrUtl from disk 35          MulsDef Scans a set of object modules and libraries searching for multiply defined symbols          MyUpdate Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process          Plot Computes and displays 3 dimensional functions in hires          Polygon Main pattern generator with color cycling          QMouse Queries whether a mouse button is pressed. This can give a return code that can customize a startup-sequence based on whether a mouse button was pressed          Touch Example of setting the timestamp on a file, using a technique from Commodore-Amiga          Trees More extensive version of the trees program on disk 31</p> <p><b>Fred Fish Disk 50</b>          Aem Version 1.1 of a shareware 68000 macro assembler, compatible with the Metacomco assembler. This includes an example startup module and more Motorola mnemonics          BreakOut A brick breakout game, uses 3-D glasses          DiskZap Version 1.1 of a program to edit disks and binary files          FirstSilicon A smart CLI replacement with full editing and recall of previous commands          Sizzlers A Missile Command-type game, with sound, in assembler          PerfectSound Sound editor for a low-cost sound digitizer          Sizzlers Graphics demos          Ver of 'tah' for Unix System V machines, in C          Wombat Version 3.01 of Dave Wecker's terminal emulator</p> <p><b>Fred Fish Disk 51</b>          Bison GNU for Unix 'yacc', working update to FF4          Compress Update to the file compression program on disk 6          Cos 'Wheel of Fortune'-type game in AmigaBasic          DrSaed Unix-like 'diff' and 'sae' for finding the differences between two files, and then reconstructing the other, given one file, and the list of differences          Portable versions of the CPM squeeze and unsqueeze</p> <p><b>Fred Fish Disk 52</b>          Assign Replacement for AmigaDOS 'assign' command in C          Fractal Makes random fractal terrains          Poly, HAMPoly Workbench-type demos for making polygons in hires and HAM          MacGads Example of mutual exclusion gadgets with GadgetText          Tek4010 Tektronix 4010 terminal emulator          VDraw Versions 1.16 and 1.18 of a Deluxe Paint-like drawing program</p> <p><b>Fred Fish Disk 53</b>          Animations Demo animations with player program for Aegis Animator          ArcArc Creates rename scripts for files with long names, so they can be easily 'arced' and 'unarced'          ARP Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'find', 'ls', 'mkdir', 'rm', 'rmdir', 'touch', 'unlink', and 'wait'          Compiler Not fully ported to the Amiga, this is a 68000 C compiler. It will produce simple assembly language output, but needs a lot of work          Spreadsheet Update with source of the 'vc' spreadsheet on disk 36          TarSplit Port of program to split Unix 'tar' archives          Utlencodes Utilities to encode and decode binary files for ASCII transmission, expanding them by 35%</p> <p><b>Fred Fish Disk 54</b>          Hanoi Solves Towers of Hanoi Problem in it's own Workbench window, by Al Czer          ISpell Port of a Unix screen oriented, interactive spelling checker. (Expansion RAM required) by Pace Willison          Ing A screen of lots of bouncing little windows by Leo 'Bois Ewhac' Schwab          Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods, by William Rucklidge          Programs to play/record through the MIDI IF, by Fred Cassier          MoreRows Program to make the Work Bench Screen larger than normal, by Neil Katin and Jim Madkraz          Tilt Program to make your Amiga look like it didn't pass vibration testing, by Leo 'Bois Ewhac' Schwab</p> <p><b>Fred Fish Disk 55</b>          Cah V2.05 of Matt Dillon's csh like shell (Modified for Marx C), by Matt Dillon          Modified by Steve Drew          NewStartups New C Startup modules          AStartup.asm with 1.2 fixes and better quote handling          TWStartup.asm opens a stdio window, using user spec, by Commodore          posted to BIX by Carolyn Schepher          Change another program's screen colors, by Carolyn Schepher          PipeDevice Allows the standard output of one process to be fed to the standard input of another, by Matt Dillon          ScreenSave Save a normal or HAM mode screen as an IFF file, by Carolyn Schepher          Shanghai Demo of the Activision game Shanghai          SoundExample A double buffered sound example for Marx C, by Jim Goodnow          Vspites A working vspite example, by Eric Cotton</p>	<p>Vt100 V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem, by Dave Wecker</p> <p><b>Fred Fish Disk 56</b>          Clipboard Clipboard device interface routines, to provide a standard interface, by Andy Finke          Demos of the use of DOS Packets, ConLink, etc, by Carolyn Schepher          Program to find all available disk device names and return them as an exec list, by Philip Lindsey          GetVolume Program to get volume name of the volume that a given file resides on, by Chuck McManis          Icon2C Reads an icon file and writes out a fragment of C code with the icon data structures, by Carolyn Schepher          Program to merge the MemLink entries of sequentially configured RAM boards, by Carolyn Schepher          An object oriented drawing program, Vt1.1 by Tim Mooney</p> <p><b>Fred Fish Disk 57</b>          Replaced by FF#7 Due to Copyright problems</p> <p><b>Fred Fish Disk 58</b>          ASDG-rnd Extremely useful shareware recoverable ram disk, by Perry Kivolowitz          Displays any IFF picture, independent of the physical display size, using hardware scroll, by John Hodgson          Reads pairs of x and y values from a list of files and draws a formatted graph, by Laurence Turner          Shareware data management system, V1.5          Walks through the free memory lists, zeroing free memory along the way, by John Hodgson</p> <p><b>NewZAP</b> A third-generation multi-purpose file sector editing utility, V3.0 by John Hodgson  <b>RainBow</b> A Maunader-style rainbow generator, by John Hodgson  <b>SMUSPlayers</b> Two SMUS plays, to play SMUS IFF music formatted files, by John Hodgson  <b>View</b> A tiny LBM viewer, by John Hodgson  <b>Wdmpup</b> JX-80 optimized workbench printer that does not use DumpRPort, by J. Hodgson</p> <p><b>Fred Fish Disk 59</b>          Browser Update to browser program on disks 18 and 34, S-E          Browser2 Another different browser program, E          Clock Clock program with fonts, colors, E          Dme Dillon text editor V1.22 for programmers, ED          DropCloth Puts pattern on Workbench backdrop, ED          DropShadow Puts shadows on Workbench windows, E-D          FixWB Similar to DropCloth, but doesn't work yet, S-D          mCAD Object-oriented drawing program, version 1.2.2. Much improved over disk 56, D.M.          Robotoff Demo of animated pointers on Workbench, S-E-D          Supermort General compounding/amortization loan calculator, E-D</p> <p><b>Fred Fish Disk 60</b>          Various shareware and freeware programs          Bitz Memory resident file viewer. Very fast, E-D          BitzFonts Makes text output faster, E-D          HandShake Terminal emulator with VT52/VT100/VT102 support, E-D          Med Mouse-driven text editor version 2.1, E-D          PrDryGen Generates printer drivers, version 1.1.5 available from author, E-D          Show Sideways-like IFF viewer, V2.1, E-D          Uedit Customizable text editor V2.0, E-D          Ueturbio Example Uedit setup macros, S-E-D</p> <p><b>Fred Fish Disk 61</b>          ATPatch Patches Transformer to work under AmigaDOS 1.2, S-E-D          FilDisk Writes zeroes to free blocks on a disk for security, S-E-D          LPatch Patch for programs that abort when loading under AmigaDOS 1.2, S-E-D          MicroEmacs Conroy MicroEmacs V3.8a, newer than disk 22, S-E-D          PearlFont Like Topaz, but rounded edges          Terrain Generates fractal scenery, S-E-D          VSpites Makes 28 Vspites, from P&amp;E&amp;B&amp;K</p> <p><b>Fred Fish Disk 62</b>          This is a port of the Unix game 'Hack', by the Software Distillery, version 1.0.3D.</p> <p><b>Fred Fish Disk 63</b>          This is a port of the Unix game 'Larn', by the Software Distillery, version 12.0B.</p> <p><b>Fred Fish Disk 64</b>          This is an official IFF specification disk from Commodore, an update to disk 16.</p> <p><b>Fred Fish Disk 65</b>          Bawk Unix text processor, like 'awk'. Doesn't work, but source is included, S-E-D          MWB Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D          CloseWB Example for closing a custom Workbench screen, S-E-D          Cookie Generates one-line fortune-cookie aphorisms, S-E-D          Build-your-own mouse port clock. Creates C source files for menus, based on text descriptions, S-E-D          CSB Tutorial on new packets and structures in AmigaDOS 1.2          PascalToC Pascal to C translator, not so great, S-E-D          'tah'-like FORTRAN preprocessor, S-E-D          Starts programs from CLI, allowing CLI window to close, E-D</p>
---	---	---	---



<p><b>SunMouse</b> This program automatically clicks in windows when the mouse is moved over them. V1.0, E-D</p> <p><b>Fred Fish Disk 66</b></p> <p><b>AmSoc</b> Preliminary plans for a SCSI disk controller board.</p> <p><b>Am58k</b> Macro assembler, version 1.0.1, E-D</p> <p><b>Assigned</b> Example for avoiding DOS insert-disk requester, by scanning the list of 'assigned' names. S-E-D</p> <p><b>Dk</b> Pretends to set away at CLI window. S-E-D</p> <p><b>Flip</b> Flips whole screen as a joke. S-E-D</p> <p><b>Foogol</b> Foogol cross-compiler generates VAX assembly code. S-E-D</p> <p><b>Free</b> Prints amount of free space on all drives. S-E-D</p> <p><b>MailocTest</b> Mailloc free memory test program. S-E-D</p> <p><b>Melt</b> Pretends to melt the screen. S-E-D</p> <p><b>Nart</b> Graphic flying string demo. S-E-D</p> <p><b>Pury</b> Easy way to set printer attributes from Workbench. E-D</p> <p><b>RayTracer</b> Simple ray tracing program. E-D</p> <p><b>SendPackets</b> Updated CBM examples of packet routines on disk 35. S-E-D</p> <p><b>SnapShot</b> Memory resident screen dump. E-D</p> <p><b>TagBBS</b> Shareware BBS system, version 1.02.</p> <p><b>Fred Fish Disk 67</b></p> <p><b>AmCat</b> Shareware disk cataloging program.</p> <p><b>AmigaSpell</b> Shareware Intuition spelling checker, V2.0. E-D</p> <p><b>Bouncer</b> 3-D bouncing ball written in MultiForth, SED</p> <p><b>Comm</b> Terminal program version 1.33, E</p> <p><b>Dux5</b> Another version of DirUtil. S-E-D</p> <p><b>HexCalc</b> Hex, octal, &amp; decimal calculator. E-D</p> <p><b>Ions</b> Various big and alternate image icons.</p> <p><b>Mandala</b> Mandala graphics and sound. E</p> <p><b>PersMat</b> Demo shareware personal file manager.</p> <p><b>RSLClock</b> Menu bar clock version 1.3. E-D</p> <p><b>RTGubes</b> Graphics demo of 3D cubes. E-D</p> <p><b>Wheel</b> "Wheel of Fortune"-type game, AmigaBASIC</p> <p><b>Fred Fish Disk 68</b></p> <p>This is version MG 1b of the MicroGNUMacs. Source and executable are included, as well as source for other computers besides the Amiga.</p> <p><b>Fred Fish Disk 69</b></p> <p><b>Am58k</b> Macro assembler, v1.0.3, E-D</p> <p><b>BitLab</b> Bitler exploring program, in C, S-E-D</p> <p><b>Connan</b> Replacement console device handler adds editing and history to any application that uses CON. v0.8, E-D</p> <p><b>Console</b> Replacement console routines, in C, S-E-D</p> <p><b>Dk</b> Decays the screen bit by bit, update to disk 66, in Module 2, S-E-D</p> <p><b>Frag</b> Displays memory fragmentation by listing the size of free memory blocks, in C, S-E-D</p> <p><b>IconType</b> Change the type of an icon, in C, S-E-D</p> <p><b>Make</b> 'make' in Manx C, S-E-D</p> <p><b>MonProc</b> Monitors processes for packet activity, in C, S-E-D</p> <p><b>MouseClock</b> Mouse pointer into a digital clock, in C, SED</p> <p><b>So</b> Browses system structures, from Transactor magazine, v1.0, in C, S-E-D</p> <p><b>Spew</b> Generates National Enquirer-type headlines from rules file, in C, S-E-D</p> <p><b>Spool</b> Three programs to demonstrate multitasking &amp; spooling in a printer spooler. In C, v1.2, S-E-D</p> <p><b>Wc</b> Counts words ala Unix 'wc', but faster, in C, S-E-D</p> <p><b>Fred Fish Disk 70</b></p> <p>This is a disk of shareware programs.</p> <p><b>AmigaMonitor</b> Explores state of the system, v1.13</p> <p><b>Arc</b> Standard file compressor and librarian, v0.23, a port of MS-DOS v5.0. E-D</p> <p><b>BlackBook</b> Phone book program.</p> <p><b>DoIt</b> Intuition-driven file manipulator program, v2.0</p> <p><b>GravityWars</b> Game of planets, ships and black holes, v1.03</p> <p><b>Jobs</b> Alternate user interface to CLI and WB, v2.1</p> <p><b>Lens</b> Magnifies area around mouse, shows it in a window, v1.0</p> <p><b>Life-3d</b> 3D version of the classic cellular-automaton game, v1.2</p> <p><b>Logo</b> Logo language interpreter</p> <p><b>SetKey</b> Demo keypad editor, v1.0</p> <p><b>Vpg</b> Makes displays for aligning video monitors, v1.0</p> <p><b>Fred Fish Disk 71</b></p> <p><b>ArFol</b> Makes airfoils using the Joukowski transformation, in C, S-E-D</p> <p><b>Amiga Basic</b> Miscellaneous programs including 3D plot program, a kaleidoscope, C-A logo drawing program file comparison utility string search program, S-E-D</p> <p><b>Blocks</b> A variation of "lines", but with variable color blocks. E-D</p> <p><b>Comm</b> Great terminal program, v1.34, E-D</p> <p><b>DiskX</b> Utility for exploring file system. E-D</p> <p><b>Fpic</b> Simple image processing program that operates on IFF pictures, with several filters, merging images, E-D</p> <p><b>IconMk</b> Makes icons for files, v1.2a, E-D</p> <p><b>Ions</b> New icons</p> <p><b>NonStars</b> Two new fonts: "helix", an electronic circuit element font, and "bm5", a PC-like font.</p> <p><b>PerCLI</b> An AmigaBASIC CLI shell program.</p> <p><b>PWDemo</b> Demo of the commercial product.</p> <p><b>PowerWindows</b> v1.2. Adds creation of custom windows, menus, and gadgets, giving C or assembly source. E-D</p> <p><b>Rot</b> Creates and animates 3-D objects, v0.5, E-D</p> <p><b>TimeSet</b> Sets time from Workbench, E-D</p> <p><b>Fred Fish Disk 72</b></p> <p>This is a disk of IFF pictures.</p> <p><b>Fred Fish Disk 73</b></p> <p><b>Add</b> Customizes existing program menus with Amiga-key shortcuts. Also includes 'unil', which waits until a given window is created. Shareware, in C, S-E-D</p>	<p><b>AutolconOpen</b> Feels WB into thinking mouse has double-clicked icons. In C, S-E-D</p> <p><b>Dio</b> Generic Exec device interface code for opening libraries, getting multiple IO channels, asynchronous operations, etc. In C, S-E-D</p> <p><b>Dissolve</b> Slowly displays IFF files, ala Nov 86 Dr. Dobbs' program. In C, S-E-D</p> <p><b>DTerm</b> Flexible, reprogrammable terminal program v1.10, E-D</p> <p><b>Expose</b> Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. In C, S-E-D</p> <p><b>Lt</b> "Long Movie", program views series of IFF pics in quick succession, up to 18 fps. Shareware, E-D</p> <p><b>Lmv</b> Mouse pointer disappears after ten seconds of non-use. In C, S-E-D</p> <p><b>MouseOff</b> Examples of controlling parallel port, with resources instead of the PAR device. In C, S-E-D</p> <p><b>ParOut</b> Child-like font.</p> <p><b>PenPalFont</b> Similar to RunBack on disk 66, runs program from the CLI allowing the CLI window to close. In C, S-E-D</p> <p><b>RunBackGround</b> Screenshot utility, update FF 66-E-D</p> <p><b>SnapShot</b> Example installs a device handler before Intuition, and speaks each key as it is pressed. In C and assembler, S-E-D</p> <p><b>TypeAndTel</b> Prints info about system lists, in assembler, S-E-D</p> <p><b>Xpor</b></p> <p><b>Fred Fish Disk 74</b></p> <p><b>Ced</b> Edits and recalls CLI commands, v1.3, E-D</p> <p><b>Control</b> Intercepts graphic printer dump calls and accesses color map, width, and screen resolution. C, S-E-D</p> <p><b>Dme</b> Simple WYSIWYG text editor for programmers, v1.25. Update of FF 59-E-D</p> <p><b>DropShadow</b> WB dropshadows, v2.0. Update FF 59-E-D</p> <p><b>Funds</b> AmigaBASIC prog tracks mutual or stocks-D</p> <p><b>Less</b> Text viewing program, like Unix 'more', v1.1, update to disk 34. S-E-D</p> <p><b>Makemake</b> Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D</p> <p><b>mCAD</b> Object-oriented drawing prog, v1.24, update to FF 58. Shareware, E-D</p> <p><b>Random</b> Simple random number generator in C. S-E-D</p> <p><b>TDebug</b> Monitors devices by intercepting Exec SendIO() and DoIO() vectors, in C, v1.0, S-E-D</p> <p><b>Units</b> Converts measurements in different units, includes 'chart' option, in C, S-E-D</p> <p><b>XCopy</b> Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D</p> <p><b>Fred Fish Disk 75</b></p> <p><b>Bezier</b> Play with Bezier curves points and granularity, S-E-D</p> <p><b>BSplines</b> Play with b-splines, as above, S-E-D</p> <p><b>Comm</b> C source for Comm terminal program v1.34. S-E-D</p> <p><b>Copy</b> Replacement 'copy' command v1.0, preserves date, in C, S-E-D</p> <p><b>Diff</b> Simple 'diff' in C, S-E-D</p> <p><b>DuM2</b> Another DirUtil in Module-2, v1.5, S-E-D</p> <p><b>Eless</b> Fast 'dir' program in C, S-E-D</p> <p><b>Fd</b> Faster 'less' in C, S-E-D</p> <p><b>HardCopy</b> Sends a transcript of a CLI session to a file, in C, S-E-D</p> <p><b>MouseOff</b> Update FF73, turns off mouse pointer, S-E-D</p> <p><b>SeFort</b> Changes the font in a Workbench screen, v2.0, S-E-D</p> <p><b>SpeedDr</b> Another fast 'dir', in assembler, S-E-D</p> <p><b>Fred Fish Disk 76 &amp; 77</b></p> <p>These are disks 1 and 2 of Chris Gray's Draco distribution for the Amiga. Draco is a compiled, structured language reminiscent of both C and Pascal. A full interface to AmigaDOS and Intuition is supplied. Be sure to get both disk 76 and 77.</p> <p><b>Fred Fish Disk 78</b></p> <p><b>Cycles</b> Cycle game like 'Tron', v1.0, E-D</p> <p><b>ECMS</b> Experts Only Mercenary Simulator game, E-D</p> <p><b>MandelVroom</b> Mandelbrot generator with enhanced palette controls, fixed/float point, presets, v1.50, in Manx C, S-E-D</p> <p><b>Fred Fish Disk 79</b></p> <p><b>AmTools</b> CLI tools in assembler: echo, loadit, mounted, setbase, why, S-E-D</p> <p><b>AssignDev</b> Give devices multiple names, in C, S-E-D</p> <p><b>AuxHandler</b> Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew</p> <p><b>Cnd</b> Redirects printer output to a file, in C, S-E-D</p> <p><b>Info</b> AmigaDOS 'info' replacement, in C and assembler, S-E-D</p> <p><b>Kill</b> Removes a task and its resources, in C, S-E-D</p> <p><b>M2Error</b> Displays errors from TDI Module-2 compiles, S-E-D</p> <p><b>MonProc</b> Update to process packet prog. from FF69, in C, S-E-D</p> <p><b>Mounted</b> Program for testing if a drive is present, in a script in C, S-E-D</p> <p><b>Nro</b> Another 'off-style' text formatter, in C, S-E-D</p> <p><b>ParTask</b> Finds parent task, in C, S-E-D</p> <p><b>QueryAny</b> For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D</p> <p><b>ScrSizer</b> Resets pref settings for screen size, in C, SED</p> <p><b>SharedLib</b> Simple CreateTask() example in C, S-E-D</p> <p><b>Taskik</b> Unix Windows client v1.0, in C, S-E-D</p> <p><b>Uw</b> Lists tasks on ready and wait queues, in C, S-E-D</p> <p><b>Who</b> Demo of the commercial product.</p> <p><b>Fred Fish Disk 80</b> (see Fred Fish 90)</p> <p><b>Fred Fish 80</b> has been withdrawn due to copyright problems.</p> <p><b>Fred Fish Disk 81</b></p> <p><b>Am58k</b> V1.1.2 of a macro assembler</p> <p><b>Am58k</b> Shrinks the FACC window and moves it to the back</p> <p><b>AutoFac</b> 53 count IFF brushes of electronic symbols</p> <p><b>Brushes</b> Checks structure of an IFF file GiodV1.4</p> <p><b>CheckFF</b> update FF74 of a simple CLI</p> <p><b>Conman</b> Replaces console handler to add editing and history to many programs</p> <p><b>Fonts</b> Miscellaneous fonts</p> <p><b>KeyLock</b> V6.0 of the Icon programming language</p> <p><b>KeyLock</b> Freezes the keyboard and mouse until pass word entered.</p>	<p><b>ScatDisplay</b> hack created from 'Ing'</p> <p><b>Smush</b> Smushes an IFF file.</p> <p><b>Target</b> Each mouse click becomes a gunshot</p> <p><b>Fred Fish Disk 82</b></p> <p><b>Adventure</b> Port of the classic Crowther and Woods game</p> <p><b>AmiTerm</b> V0.50 of a telecommunications program, with scripts, redial, beeps, enhanced file requester</p> <p><b>D2D-Demo</b> Demo version of Disk-2-Disk by Central Coast Software</p> <p><b>DX-Synth</b> Voice file program for Yamaha DX series synthesizers, update to disk 38</p> <p><b>DiskMan</b> V1.0 of another DirUtil program</p> <p><b>Panel</b> Miscellaneous new icons</p> <p><b>Rocket</b> Universal MIDI patch panel, v1.2</p> <p><b>Sand</b> Another Workbench hack, plays Lunar Lander</p> <p><b>Sand</b> Game of sands following your pointer.</p> <p><b>Fred Fish Disk 83</b></p> <p>This disk contains a demo version of TeX from N. Squared.</p> <p>It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided.</p> <p><b>Fred Fish Disk 84</b></p> <p><b>AudioTools</b> Programs from Rob Peck's July/August Amiga World article</p> <p><b>BitLab</b> Bitler experimentation program, v1.2, update to FF69</p> <p><b>Ed</b> Simple editor, similar to Unix 'ed', based on the editor in Software Tools.</p> <p><b>GravityWars</b> Game of planets, ships and black holes, v1.04, update to disk 70.</p> <p><b>HunkPad</b> Adds legal padding to executables for Xmodem transmission.</p> <p><b>PipeHandler</b> An AmigaDOS pipe device which supports named pipes and taps. V1.2</p> <p><b>PopCLI</b> V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40.</p> <p><b>Requester</b> Update FF34, file requester similar to DCPaint</p> <p><b>ScotDevice</b> V3.1 of a mountable MicroForge SCSI driver.</p> <p><b>Vicom</b> Another screen hack, makes TV-like static on Schwarz Parody</p> <p><b>Fred Fish Disk 85</b></p> <p><b>Csh</b> V2.06 of Dillon's 'csh'-like shell</p> <p><b>FileReq</b> Source to wildcard file requester</p> <p><b>Hide</b> Hides expansion memory from programs</p> <p><b>ImageTools</b> Shareware tools to manipulate IFF images</p> <p><b>LowMem</b> ServerShared library to aid in low memory situations</p> <p><b>Plot</b> A star plotting program with source.</p> <p><b>RawD</b> Example of setting raw mode on standard input</p> <p><b>Rocket</b> Lunar Lander for Workbench, with source.</p> <p><b>Vmore</b> 'more'-like text viewing utility, v1.0 SE</p> <p><b>Vnews</b> Simple Unix news reader.</p> <p><b>Fred Fish Disk 86</b></p> <p><b>AutoPoint</b> Auto-selects window under the mouse pointer, with screenwaver.</p> <p><b>ClickToFront</b> Double-clicks in window brings it to front, v1.1, S-E-D</p> <p><b>CndT</b> V3.0 of a tool to redirect printer output to file.</p> <p><b>FileISG-Demo</b> Demo of Software File Ilog, a database manager with sound and graphics.</p> <p><b>Fred Fish Disk 87</b></p> <p><b>Adventure</b> system from Byte May 1987, v1.2 E-D</p> <p><b>AdvSys</b> Foals Workbench to open disk icons, V1.2</p> <p><b>AutoConOpen</b> update to disk 73, S-E-D</p> <p><b>Ciaz</b> Converts IFF files to PostScript, V2.0, SED</p> <p><b>Commodi</b> teAmicaz's Commodities Exchange, an exec library to manage input handler, v0.4</p> <p><b>Diff</b> Update to disk 75 of Unix-like 'diff', S-E-D</p> <p><b>Dme</b> V1.27 of Dillon's text editor, update FF74-E-D</p> <p><b>DropShadow</b> V2.0 of prog. that puts shadows on Workbench, S-E-D</p> <p><b>Elb</b> Shared library example in Manx C.</p> <p><b>Elb-Handler</b> An AmigaDOS device handler generates unique identifiers, V1.0, S-E-D</p> <p><b>Install</b> Alternate AmigaDOS 'install' programs, SED</p> <p><b>MemWatch</b> Waits for low memory trashing, V2.0, SED</p> <p><b>MovePointer</b> Moves pointer to given location, S-E-D</p> <p><b>MoveWindow</b> Move window to given location, S-E-D</p> <p><b>MunchingSq</b> Munching Squares hack, S-E-D</p> <p><b>PatTest</b> Test to see if this is a PAL machine, S-E-D</p> <p><b>Sc</b> Generates random scenery, S-E-D</p> <p><b>Tek4695</b> Tek4695 printer driver</p> <p><b>WBOUPFF</b> Example of dual-playfield screen, update FF41, S-E-D</p> <p><b>WarpText</b> Fast text rendering routines, S-E-D</p> <p><b>Yaff</b> Example IFF reader, S-E-D</p> <p><b>Zoo</b> A file archiver like 'arc', v1.42A, E-D</p> <p><b>Fred Fish Disk 88</b> (see Fred Fish 80)</p> <p>FF Disk 88 has been removed due to copyright problems</p> <p><b>Fred Fish Disk 89</b> (replaces Fred Fish 80)</p> <p><b>DiskMaster</b> Disk catalogue program, V1.0a, E-D</p> <p><b>FuncKey</b> Shareware function key editor, V1.01, E-D</p> <p><b>MFF-Demo</b> Demo of MicroFiche Filer database prog</p> <p><b>ScreenShift</b> Adjust screen position in Preferences.SED</p> <p><b>Snake</b> Bouncing squiggly lines demo, S-E-D</p> <p><b>AutoEngineer</b> screen caption requester improvement S-E-D</p> <p><b>Demolition</b> Display Hack S-E-D</p> <p><b>Fred Fish Disk 89</b> (replaces Fred Fish 80)</p> <p><b>AmiGazer</b> Night sky viewer of 1573 stars, set date, time, day. E-D</p> <p><b>CardFile</b> AmigaBASIC and file study aid. E-D</p> <p><b>Comman</b> Console handler replacement gives line editing and history to most prog, v0.98, ED</p> <p><b>MandelVroom</b> Slight update to disk 78 Mandelbrot program, E-D</p> <p><b>NewDemos</b> Replacements for lines and boxes demos that take less CPU time, E-D</p> <p><b>Othello</b> Game of Othello, E-D</p> <p><b>PrintText</b> Displays text files with gadgets, speech, IFF display, v1.2, E-D</p> <p><b>PrbDrvGen</b> Automatic printer drv. generator v2.2a, ED</p> <p><b>RainBench</b> Oxydes colors of WB backdrop or text. ED</p> <p><b>ShortCut</b> Makes single-key shortcuts for entering commonly typed CLI commands &amp; custom macros. E-D</p> <p><b>ShowPrint</b> Displays and prints all sizes of IFF pictures &amp; controls printer output styles, v2.0 E-D</p> <p><b>Sizzlers</b> Graphics demos, v1.7.0, E-D</p> <p><b>Timer</b> Small Workbench timer counts time and \$/minute, E-D</p> <p><b>Tools</b> Intronics tools: a memory editor, memory disassembler, ASCII chart, and calculator. E</p>	<p><b>Fred Fish Disk 91</b></p> <p><b>Adventure</b> Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostanick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunniff. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Lattice 3.03. CLI environment only. Documentation is available from the authors.</p> <p><b>Fred Fish Disk 92</b></p> <p><b>As6502</b> portable 6502 assembler, C source, by J. Van Ornum, Amiga port by Joel Swank</p> <p><b>Bawk</b> Text processor update from FF65 inspired by UNIX awk. Searches files for patterns, performs actions based on patterns. By Bob Brodt; Amiga port by Johan Widen</p> <p><b>HunkPad</b> update of FF84 version, by J. Hamilton, pads an object file to a multiple of 128 bytes for better xmodem transfer. S E</p> <p><b>Less</b> Like Unix 'more', better, version 1.2 update of FF74. Scrolls back and forward. S E by Mark Nudelman, Amiga port by Bob Leivan.</p> <p><b>Ndr</b> Library that implements the ABSO Unix dir access routines by Mike Meyer. S</p> <p><b>Parse</b> Recursive descent expression parser, computes, and prints expressions, includes transcendental function support. C source, included by J. Olsen</p> <p><b>Shar</b> Two programs to pack and unpack shell archives includes C source, by Fabian G. Dufosse</p> <p><b>SmallLib</b> 8 times smaller AmigaLib replacement, binary only, by Bryce Nesbitt</p> <p><b>UUnencode</b> Encodes/decodes binary files for e mail or text-only methods. Update of FF53, includes checksum technique, compatible with older versions, plus transparent to older versions options. By Mark Horton, modified by Alan Rosenthal and Bryce Nesbitt.</p> <p><b>Fred Fish Disk 93</b></p> <p><b>Dme</b> Version 1.27 WYSIWYG programmer editor. Not a word processor. Includes key mapping, text scrolling, line-style statistics, multiple windows, ability to iconify windows. Update of FF87, includes source code, by Matt Dillon</p> <p><b>MicroEmacs</b> Version 3.8i, update to FF61 includes source. Orig by Dave Conroy multiple modifications by Daniel Lawrence</p> <p><b>Fred Fish Disk 94</b></p> <p><b>AudioTools</b> Demo programs from Rob Peck's July/August issue of AmigaWorld on accessing the audio device.</p> <p><b>V2</b> update of FF84. S, by Rob Peck</p> <p><b>ClickToFront</b> Similar in function to ClickToFront prog (FF86), bring windows to front by clicking on any part of them. V1.0. by Davide Cervone SE</p> <p><b>HeliosMouse</b> Automatically activates a window simply by moving the mouse pointer into the window. V1.0. Includes source. By Davide Cervone</p> <p><b>IFF2Ps</b> Convert any IFF file to postscript for printing or viewing on a postscript compatible device. Version 1.2, by William Mason and Sam Paulucci E</p> <p><b>ModuleTools</b> Various Module 2 programming routines. by Jerry Mack</p> <p><b>Terrain3D</b> Pseudo-random 3d relief scenery generator, update of 'arc', FF87, by Chris Gray, 3d by Howard Hull</p> <p><b>Fred Fish Disk 95</b></p> <p><b>Cnd</b> redirects the serial device or parallel device output to a file. Capture print jobs, debug or 'offline' printing. V4. By C. Scheppner S</p> <p><b>OygnusEdDemo</b> Demo of OygnusSoft's OygnusEd editor, a multiplexed, multiple feature editor. Includes demo 3.0 of MandXP. By OygnusSoft Software E</p> <p><b>Gom!</b> "Get Out My Face" makes the Guru go away to allow clean-up &amp; shutdown more cleanly. V1.0, by Christian Johnsen E</p> <p><b>Journal</b> records sequence of mouse &amp; keyboard events, stored in a file for future playback. Good for demos or documenting bugs. E. by D. Cervone</p> <p><b>Mergemem</b> attempts merging of MemList entries of sequentially configured ram boards. When successful, allows allocating a section of memory which spans both boards. V2, update of FF56, by Carolyn Schnepfer SE</p> <p><b>PrinterStealer</b> Asimilar to "Cnd", allows diversion of output destined for printer to a file. Binary only. Source avail. from authors. By A. Livshits &amp; J-M Forges</p> <p><b>Record-Replay</b> similar to 'Journal', records and plays back mouse and keyboard events. B only. source avail. from authors, Alex Livshits &amp; J-M Forges</p> <p><b>Fred Fish Disk 96</b></p> <p><b>AnimPlayer</b> Animation reader and display by the combined efforts of Videocap, Sculpt3D, Silver, Forms-In-Flight, and Animator Approximate M. Hasehal.</p> <p><b>Chess</b> Amiga port of Jan-Atanga interface. High playability. V1.0. S, by J. Amiga, Amiga port by B. Leivan</p> <p><b>Hadbench</b> provides source for WB-Likeprog, for experimentation &amp; validation of new interface ideas. Next a WB replacement. by Bill Koversey</p> <p><b>Label</b> Print labels with arbitrary text. V1.3, Source available from author, M. Hansen</p> <p><b>LineDrawer</b> Produces line drawing based on drawing commands stored in a text file. Includes demo that draws an outline map of the USA and state borders. V1.0, SE, by John Olson</p> <p><b>PopUpMenu</b> Example code implementing pop-up menus, reasonably compatible with Intuition menu SE, by Derek Zahn</p> <p><b>Tek4695</b> Tektronix 4695/4696 printer driver. SE, by P. Staud</p> <p><b>TimeRan</b> Fast and Chip ram test prog. E by Takashi</p> <p><b>WarpText</b> Fast text rendering routines, to be linked with application prog. Test display 'as fast or faster than "bitz"'. V2.0 update of FF87, S by Bill Kelly</p> <p><b>Fred Fish Disk 97</b></p> <p><b>Replaces</b> FF57 for Copyright problems</p> <p><b>OutAndPaste</b> Implementations of Unix cut and paste commands. by John Weald</p>
---	---	---	---



Graphit	Program to plot simple functions in 2 or 3 dimensions. by Flynn Fishman	QuickFix	An FF slideshow and cell animation prog. v1.13.	Marketroid	Another devious sprite oriented demo with lots of "in" jokes. 512K required. Includes S. By Leo Schwab	Car	This animation is one of Allen's entries to the Badge Killer Demo Contest. It appears to be an inside joke relating to a well known Amiga's experience with a certain high-end graphics hardware manufacturer.
Juggler	V1.2 of robot juggler animation. Uses HAM mode and ray tracing. by Eric Graham	Risolvola	A Finnish game. Also called Go-Moku. v1.0	Fred Fish Disk 116	Another devious sprite oriented demo with lots of "in" jokes. 512K required. Includes S. By Leo Schwab	Fred Fish Disk 124	Some sample animated icons. You might find just the icon for that elusive CLI program you've been meaning to make run from the Workbench environment.
MouseReader	Shareware program to read text files & view IFF files using only the mouse. by William Betz	Diff	A utility similar to other common "diff" programs. S. Suite provides ex. code of facilities such as FileIO Requester, XText, OutRequest, & tutorial on how to program the Amiga. Book 1.01.5	ExP_Demo	Demo version of Express Paint 1.1. Used to create the scrolling demo picture in the AMUC_Demo drawer on the disk. B only. By Stephen Vermeulen	HBHill	This animation is one of Kevin's entries to the Badge Killer Demo Contest. It is the first known animation that makes use of the Amiga's "Extra Half Byte" mode.
Splines	Prog to demonstrate curve fitting & re-rendering techniques. by Helene (Lee) Taran	ProSuite	A utility to send common control settings to PRIT. device. S.	Stars	Based on original code by Leo Schwab, has credits longer than the actual demo. Runs on 512K Amiga. B only. By Hobbie Onis	Iconify	A subroutine that creates an icon on the Amiga screen that can be subsequently dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. With source and demo program. By Leo Schwab
Shm	Simple graphics demo, approximately simulates the motion of two interacting pendulums. Includes Source by Chris Edies	SVTools	A utility to send common control settings to PRIT. device. S.	WireDemo	Demonstrates the Amiga's line drawing speed. Runs on a 512K Amiga. Includes S. By Matt Dillon	OnlyAmiga	This animation is John's entry to the Badge Killer Demo Contest. It consists of five balls being juggled by pyramids rotating on their tops. By John Singh Hans
Fred Fish Disk 88	16 color terminal program based on Comm V1.34. Includes Macro window, custom gadgets, colored menus, etc. V. Beta 0.18 by Keith Young. comm by D.J. James. E	Fred Fish Disk 108	Dr listing prog. based on LDR prog S	Fred Fish Disk 118	This is a complete rewrite, from the ground up, in Draco, of Peter Langston's Empire game. A multiplayer game of exploration, economics, war, etc. can last months. Played either by local keyboard or through modem. V1.0, shareware, & includes S code. By Chris Gray, original game by Peter Langston	Supb	The support library needed to rebuild various programs of Matt's from the sources, including DME, DTERF, etc. Includes source. Author: Matt Dillon
Access	16 color terminal program based on Comm V1.34. Includes Macro window, custom gadgets, colored menus, etc. V. Beta 0.18 by Keith Young. comm by D.J. James. E	MoniDCMP	Lets you monitor the IntraMessages that pass through an IDCMP window. Prints the message class, mouse coordinates, qualifier values. Great for debugging. S.	HAMmm	Displays lines whose end points are bouncing around the screen, which is a double buffered HAM screen. The Y positions of the points are continuously copied into an audio waveform that is played on all four channels, & the pitch of a just intoned chord is derived from the average X position of these points. JForth. Source By Phil Burk	VCheck	Version 1.2 of the virus detection program from Commodore Amiga Technical Support. This version will test for the presence of a virus in memory, or on specific disks. Binary only. Author: Bill Koester
Backup	Writes AmigaDOS disks as the backup destination, recover files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE	PrintPop	A utility to send common control settings to PRIT. device. S.	Fred Fish Disk 119	A 68000 assembler written in C. S.	Fred Fish Disk 127	This program is Steve and Tom's entry to the Badge Killer Demo Contest. It creates little dots that bounce around and multiply. Includes source. Author: Steve Hansel and Tom Hansel
DCDemo	DiskCat 2.3, a disk catalog program, demo limited to cataloging 100 files at a time. by Ed Alford, MicroAce Software	Sectrama	Utilities to recover lost or damaged data from floppies & hard disks. v1.1, an update to FF102.	ASB	An optimizing C compiler for the 68000 processor. update to FF53, but not based on the code of that disk.	Nemesis	This demo is Mark's entry to the Badge Killer Demo Contest. It is quite small for what it does, and won fifth place in the contest. Binary only. By Mark Riffel
HdDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SE	Tek	V100 emulator for a Tektronix 4010/4014. (V2.6) update to FF52. S.	Pdc	An optimizing C compiler for the 68000 processor. update to FF53, but not based on the code of that disk.	Ripples	This animation is one of Allen Hastings' entries to the Badge Killer Demo Contest. Unlike most other animations, it shows a fixed object from a moving point of view, rather than a moving object from a fixed point of view. Author: Allen Hastings
Obase	QuickBase, a "MailBase" Management utility, define and maintain a maximum of 200 records per file. by Kevin Hansel E	Zoo	File archiver, like "tar", v1.24B. update to FF87	Fred Fish Disk 110	A 68000 assembler written in C. S.	Fred Fish Disk 128	A 68000 disassembler, written in 68000 assembler. Includes source. Author: Greg Lee
Thal	Thal language quiz program. Speak or type english/Thai sentences from supplied file. by Alan Kent SE	Fred Fish Disk 109	Machine. A new animation.	ASB	A 68000 assembler written in C. S.	DropCloth	lets you place a pattern, a 2-bit plane IFF image or a combination of a pattern and image, into the Workbench backdrop. Version 2.2, shareware, binary only. Author: Eric Lervick
Fred Fish Disk 89	A-Render Version 3.1 Ray-Tracing Construction Set for the Amiga Computer by Brian Reed ED	SimCPM	A CPM simulator. Simulates 8080 along with h19 emulation. S.	ASB	A 68000 assembler written in C. S.	MRBackup	A hard disk backup utility, that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface & file compression. Version 1.3, includes source. Author: Mark Riffel
Fred Fish Disk 100	Must see animation, by Leo Schwab	UUpC	Hook up your Amiga as a user node. S.	ASB	A 68000 assembler written in C. S.	Paint	A simple screen painting program, written in web. Requires web preprocessing program to rebuild from source. Includes source in web. Author: Greg Lee
Benchmark	Console hardware replacement, provides line editing and command line histories transparent to application prog uses CON; displays shareware v1.0 by W. Haves. E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	PrDriver	A printer driver for the Toshiba "3" in one" printer in its Qume (best) mode. Includes source in C and assembler. Author: Rico Mariani
WBlander	Workbench display handler, upgrade of "Rocker" on FF85, now with sound effects. By Peter da Silva. E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	SOBackup	A hard disk backup utility. CLI interface only. Does file compression. Version 1.1, binary only. By Steve Drew
Fred Fish Disk 101	Circular plane generator for VideoScape3D. Generates a disklike circular polygon with the specified number of vertices. V1.0 by T. Flynn SE	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Sed	A done of the Unix & (Sed) Editor program. Includes source. Author: Eric Raymond
OPPlane	Circle plane generator for VideoScape3D. Generates a disklike circular polygon with the specified number of vertices. V1.0 by T. Flynn SE	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Keys	A "hotkeys" program that binds keyboard function keys to window manipulation functions (window activation, front to back, moving screens, etc.). Includes source. Author: Davide Cervone
IconAssembler	Change Workbench icons with IFF-brush files by Stefan Lindahl E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Fred Fish Disk 129	A pair of programs which allow you to save files, or a group of files, to one or more floppies for quick loading, does not store files in DOS format, which is why it is faster. V2.0, update to FF103. By F. Shaw, Shareware. Author: Gary Kemper
Microspell	Standalone spelling checker scans text files and reports errors. 1000 common word list, 43,000 word main dictionary with multiple user dictionary support. Interfaces with MicroEMACS 3.9 with an emacs macro to step through the source file, stopping at suspect words and allowing the user to opt-out. V1.0 by Daniel Lawrence, SED	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	MRBackup	A hard disk backup utility, does a file by file copy to standard AmigaDOS floppy disks. Includes intuition interface & file compression. Versions 2.0 (with source) and 2.1 (binary only, source available from author). Update of FF128. By Mark Riffel
Mid	mid library and utility set. Includes Midi monitor, routing utility, status utility, and more. by Bill Barton SED	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Printlet	HP Printlet printer driver. from HP sources
Pshtrp	Postscript interpreter reads and previews files on screen. by Greg Lee S(sasy)E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	PushOver	A neat little board strategy game, in AmigaBASIC. Push your pieces onto the board until you get five in a row in any direction. Includes source. Author: R.Yost
StartUp	Three C startup file replacements for standard AtariStartUp, LStartup, & LStartup. Options include (1) BothStartUp.obj, for the Workbench programs or CLI programs with or without command line parameters. (2) WBSStartUp.obj, for Workbench programs or CLI programs that require no command line parameters. (3) CLISStartUp.obj, for CLI programs that require command line parameters but do not need to be Workbench runnable. by Bryce Nesbitt SE	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	PuzzlePro	Create a puzzle from an IFF picture, which the user can then piece back together again. Written in AmigaBASIC. Version 1.0, binary only, shareware, source available from author. By Syd Bolton
Fred Fish Disk 102	Machine independent macro based C de-bugging package. Update FF41, by F. Fish profiling support by Binayak Banerjee SE	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Fred Fish Disk 120	A pair of programs which allow you to save files, or a group of files, to one or more floppies for quick loading, does not store files in DOS format, which is why it is faster. V2.0, update to FF103. By F. Shaw, Shareware. Author: Gary Kemper
Doug	Machine independent macro based C de-bugging package. Update FF41, by F. Fish profiling support by Binayak Banerjee SE	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	MRBackup	A hard disk backup utility, does a file by file copy to standard AmigaDOS floppy disks. Includes intuition interface & file compression. Versions 2.0 (with source) and 2.1 (binary only, source available from author). Update of FF128. By Mark Riffel
Match-stuff	Heavy duty text pattern matching stuff. Includes simple match text replacement capability. By Pete Goodave	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	PushOver	A neat little board strategy game, in AmigaBASIC. Push your pieces onto the board until you get five in a row in any direction. Includes source. Author: R.Yost
Sectrama	Recover lost or damaged data from floppy or hard disks or repair a damaged volume. by David Joiner E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	PuzzlePro	Create a puzzle from an IFF picture, which the user can then piece back together again. Written in AmigaBASIC. Version 1.0, binary only, shareware, source available from author. By Syd Bolton
SilCon	Simple input line interpreter with window for full editing. Upgrade FF50 by P. Goodave, E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Fred Fish Disk 121	An AmigaBASIC program that helps to convert programs written in other forms of Basic to AmigaBASIC. Author: Greg Trepel
Xcon	Use icons to call up scripts containing CLI commands. V2.0 upgrade of FF31 by Pete Goodave E	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Rot	A shareware 3-D graphics program written in AmigaBASIC, with some sample output plots. Source available from author. Author: George Trepel
Fred Fish Disk 103	Library and test prog. implement routines for creating and using trees held in memory. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Stairs	This AmigaBASIC program demonstrates a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidal relationship to their frequency. Author: Gary Coo
Ar/Trees	Library and test prog. implement routines for creating and using trees held in memory. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Uedit	Version 2.3 of this nice shareware editor. Has learn mode, a command language, menu customization, and other user configurability and customizability features. Binary only, shareware, update to version on disk 50. Author: Rick Stiles
Cac	A C cross ref. prog. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	WBColors	A simple little program to change the Workbench colors to a predetermined color set, for programs that expect to be booted off their distribution disk but instead are run from a hard disk. Includes source. Author: Stefan Lindahl
Doa/Link	A pair of progs. allows you to save files to one or more floppies for quick loading. Doesn't store Dos format. A prog. to improve control and handling of the material on all disks in "CLI-area".	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	Fred Fish Disk 122	Nicely done "blast the asteroids" type game. Unique feature is that all the images and sounds are replaceable by the end user. So instead of ships and rocks, you can have an Amiga against a horde of IBMPC's if you wish. Author: Rico Mariani
IntiDos	A prog. to improve control and handling of the material on all disks in "CLI-area".	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
MFFUpdate	A text import/utl. for MicroFiche Filer (demo on FF 89) and updates to some PD disk library databases.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Pack-It	Takes all files the files and dirs. on a disk & packs them into a single file. for modem.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Sol	Amiga version of solitaire.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Fred Fish Disk 104	Amiga version of solitaire.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
AmProgs	Misc. assembly tools. Includes some S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
BasicProgs	LeastSquare solves least square probe & graphs results. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Bson	A replacement for unix "yacc" command. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Mouse	Another prog in the tradition of command hacks. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
FlamKey	Allows keyboard and mouse inputs to be locked until a password is entered.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
GravityWars	Game of planets, ships and black holes. v2.0 update to FF84.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Po2C	A util. to write a C-lang definition to mimic the intuition pointer S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Pete-et-Fil	Ex. of creating & using reentrant processes. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Record Replay	Similar to "Journal" v2.0 update to FF95.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Fred Fish Disk 106	Shareware function key editor, v1.1, update to FF89.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
Funkey	Shareware function key editor, v1.1, update to FF89.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.
MoreArt	A small selection of some Amiga artwork.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.	ASB	A 68000 assembler written in C. S.



PatEdit	A pattern editor for creating patterns to input to the Amiga SetAPL macro call. This call sets the area fill pattern for the area filling graphics (RexFill, AreaDraw, etc.). Includes source. By Don Hyde.
OMan	Mandelbrot generator written partially in assembler for speed. Includes source. By Steve Bonner.
Fred Fish Disk 131	
Dic	Copies disks like Mauder, but multi-task. Replaces diskcopy and format (smaller than either). Includes interface. Includes source. Author: Tomas Rokicki.
HyperBase	Shareware database management system. V1.6, binary only, source available from authors. FF58 update. By: Michael MacKenzie, Marc Mengel, & Craig Norberg.
Life	A new version of Tomas's ancient Life game, with a new macro language for setting up patterns, good examples. Includes source. Author: Tomas Rokicki.
Mackie	A Popdi replacement that draws pretty lines on the screen in blanking mode. Includes source. Author: Software Designer, enhancements by Tomas Rokicki.
Mg1b	A version of Mg1 with an ARexx port and other improvements by Tomas Rokicki. Define macros & bind them to function keys in startup file. Includes source. Author: Various, enhancements by Rokicki.
WFrags	Another version of Frags, this pops up a little window that updates occasionally. Necessary for developers who wonder what their program is doing to memory. Includes source. Author: Tomas Rokicki.
Fred Fish Disk 132	
Berserk	Animation, a "must see" for every Amiga user, and ranks with "Juggler" as a premier demo for the Amiga. The difference between this distribution, and FF100, this one includes "source", use it as an example for creating animations. Fred Fish felt it was appropriate to have at least one animation that was available at the "source code" level. Author: Leo Schwab.
Fred Fish Disk 133	
Comm	Shareware replacement for the standard console handler, provides line editing and command line histories completely transparent to any application program that uses CON. windows. V1.1, binary only, updated FF100. New features include additional editing keys, fast search keys, undo key, clear history command, and more. Author: William Hayes.
Orc	Two programs useful for generating 16-bit CRC listings of the contents of disks, and verifying that a given disk's files still compute to the same CRC's as listed. V1.0, binary only. Author: Don Kindred.
OrcLists	Complete CRC check files for FF1-128 using the Orc program included on this disk. These were made directly from Fred's master disks. Author: Fred Fish.
OverScan	Patches the Intuition library so that sizable windows with MaxHeight of 200 (400 in interface) and screens with Height of 200 (400 in interface) will take advantage of the PAL overscan capability of Intuition V1.2. Useful only for European users who wish to run software written for the US market, without modifying the applications, but still using the additional space. Includes source. Author: An Freund.
Fred Fish Disk 134	
BongThroes	50 frame HAM animation done with Sculpt3D, and DipPaint. The animation took about 325 hours of runtime to generate. By Marvin Landis.
Brower	Workbench tool, using text-only windows, makes all files in the system accessible for copying, moving, renaming, deleting, etc. Billed as a "programmers workbench". V1.2, binary only. BY Peter da Silva.
Dne	V1.28 of Matt's text editor. Simple WYSIWYG editor designed for programmers. Arbitrary key mapping, fast scrolling, title-line statistics multiple windows, & ability to copy windows. FF113 update, includes source. By Matt Dillon.
Find	Utility searches for files that satisfy a given boolean expression of attributes, starting from a root pathname and searching recursively down through the hierarchy of the file system. Like the Unix find program. V1.0, includes source. By Rodney Lewis.
Library	Demo version of a shareware program that stores textual information without regard to structure or content, and allows complicated searching for specific patterns. Written in assembler for speed, binary only. Author: Bill Brownson.
SmartIcon	Shareware Intuition objects iconifier. V1.0 is limited to iconifying windows, adds a new "iconify gadget" to each window, that when clicked, confines the window into an icon in the ram:disk. Binary only, source available from author. Author: Gauthier Gault.
Fred Fish Disk 135	
TeXF	A selection of 78 TeX fonts, with a conversion program to convert them to Amiga fonts. 22 different fonts at various sizes, ranging from 15 pixels high to more than 150 pixels. Conversion program can also be used with the fonts distributed with AmigaTeX, yielding an additional 1000+ fonts for use with other Amiga programs. V2.5, binary only. By: Al Ozer.
Fred Fish Disk 136	
AsmToolBox	Assembler "toolbox" created to make interfacing between assembler programs and AmigaDOS easy. With source. By Warren Ring.
Bison	A replacement for Unix yacc command. From the GNU (GNU is Not Unix) effort. Port of the latest GNU version, by William Lohus, with the goal of preserving all of bison's current features. Includes source & test prog. "calc". By: Bob Corbett and Rich Stallman.
H2Pcs	Interactive puzzle prog. takes any FF file containing up to 16 colors, and breaks it into squares to make a puzzle the user can then piece back together again. V1.1, update of FF122, includes source. By: Al Ozer.
Paste	Version of the Unix paste utility. Paste concatenates corresponding lines of the specified files into a single output line (horizontal or parallel merging) or concatenates them into alternate lines (vertical or serial merging). Includes source. By: David Threlk
YaBong!	A game program demonstrating hardware sprite usage, including collision detection. Update of FF36. Includes source. Author: Al Ozer, based on original by Leo Schwab.
Zoo	File archiver, much like "tar" in concept, but different in implementation and user interface details. Includes features that "tar" lacks (such as filepaths names up to 255 characters in length). V1.71, update of FF108. Binary only. Author: Rahul Dhesi, Amiga port by Brian Waters.
Fred Fish Disk 137	
Ct	Program to display images from a CT scanner, along with several interesting sample images of scans of real people, including a skull, brain, heart, and spine. Each image is 256 by 256 pixels in 2048 gray scale. The display software, though it has a primitive user interface, is quite powerful, including functions like convolutions, averaging, laplacians, unsharp masking, edge detection, gradients, etc. Binary only. Author: Jonathan Harman.
JeebsIcons	Miscellaneous cute icons created for AMUC's monthly newsletter disk. Submitted by Stephen Vermeulen. Author: Steve Jeebs.
Muncho	A cute little program which plays a digitized sound sample when you insert or remove a disk from your drive. If you don't like the sounds, you can replace them with your own. Binary only. By: Andrew Werth.
St	Update to the Set Icon Type prog. on FF107. V1.10, includes source. Author: Stephen Vermeulen.
VGad	A new gadget editor that takes two pictures of the window and its gadgets, one being the normal gadget state and the other being the Lily selected state, then merges the data and converts to C source code. V1.0, binary only. Author: Stephen Vermeulen.
VirusX	A boot sector virus check program that runs in the background and automatically checks all inserted disks for a nonstandard boot sector. Such disks can optionally have their boot sector rewritten to remove the virus. Includes source. Author: Steve Tibbett.
VLabel	Program to print fancy customized disk labels. Combines an FF picture and up to 50 lines of text (which may be placed arbitrarily in any font or point size) then print the result. The FF picture can be virtually any size (up to 1008 by 1000). It will also print labels from a batch file produced by SuperBase. V1.20, binary only. By: Stephen Vermeulen.
Fred Fish Disk 138	
AmigaLine	A series of various technical notes for Amiga programmers. Author: Bryce Nesbitt.
Dff	Program that uses the same algorithm as the Unix diff program and also produces context diffs, suitable for use with patch. Binary only. Author: Unknown (Docus C diff).
Foreach	A simple but useful program that expands a wild card file specification and then invokes the specified command and once per expanded filename, with the expanded filename as the command argument. Includes source. Author: Jonas Flygare.
MacFont	A conversion tool to convert Mac fonts to Amiga fonts. Binary only. Author: John O'Neill and Rocio Mariani.
ModulaTools	Various useful routines for those programming in Modula on the Amiga. Update to version on disk 94, includes source. Author: Jerry Mack.
Vt100	Two new versions of Dave's vt100 terminal emulator. One version, based on vt100 2.6, has been enhanced by John Bersinger to include an iconify feature, add full 132 column support using overscan, and other features (binary only). The second version is release 2.8 of the main-stream version of vt100, as enhanced and supported by Tony Sumrall. Includes source. By: Dave Weicker.
Fred Fish Disk 139	
AmiGron	An enhanced and debugged version of AmiGron 2.3 from FF113. Includes source. Author: Steve Sampson, Rich Schaeffer, Christian Balzer.
ListScanner	A nice little utility to display all the Exec lists. Similar to Xplor utility FF73. Includes source in assembler. Author: Heiko Rath.
ProCalc	Simulates HP-11C programmable calculator. Both English & German versions. Shareware, binary only. Author: Gotz Muller.
RemLib	Removes a specified library (if currently unused) or displays some info on all available libraries. Includes source in assembler. Author: Heiko Rath.
TurboBackup	A fast mass floppy disk duplicator with enforced verify mode to prevent errors. V1.0, binary only. Author: Stefan Stampel and Martin Kopp.
WAngrer	Sends a window, identified by its name, to the front or back, without selecting it. Useful with AmiGron. Works on all screens. Includes source in assembler. Author: Heiko Rath.
WheelChairSim	A wheelchair simulator developed as a project for the Technical Resource Centre and the Albert Children's Hospital, to allow the matching of a wheelchair joystick to a child's handicap and allow the child to practice using the chair in a safe (simulated) environment. Binary only. Author: Unknown, submitted by Dr. Mike Smith.
Fred Fish Disk 140	
SBProlog	Volume 1 of the 2 volume Story Brook Prolog (SBP) distribution, V2.3.2. This volume contains the executables and libraries. Volume 2, on FF141, contains the C and Prolog source code. Authors: Logic Programming Group at SUNY, Stony Brook. Amiga port by David Roch and Scott Evernden.
Fred Fish Disk 141	
SBProlog	Volume 2 of the 2 volume Story Brook Prolog (SBP) distribution, version 2.3.2. This volume contains the C and Prolog source code. Volume 1, on FF140, contains the Logic Programming Group at SUNY, Stony Brook. Amiga port by David Roch and Scott Evernden.
SmallC	An Amiga port of the Small-C compiler, written by Ron Cain and published in Dr. Dobbs's Journal, in about 1980. Small-C is a rather small subset of the full 'C' language. However, it is capable of compiling itself, and other small, useful programs. Requires an assembler and linker to complete the package and produce working executables. Includes source and binary. Author: Ron Cain. Amiga port by Will Kusche.

## MOVING?



## SUBSCRIPTION PROBLEMS?

Please don't forget to let us know.  
If you are having a problem with your subscription or if you are planning to move, please write to:

**Amazing Computing Subscription Questions**  
PIM Publications, Inc.  
P.O. Box 869  
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

### Fred Fish Disk 142

Dff

Program uses same algorithm as Unix diff prog. and produces context diffs, suitable for use with patch. Same as FF138, but now includes the missing files (including source code). Author: Unknown (Docus C diff).

### Fred Fish Disk 143

FracGen

Generates fractal pictures from "seeds" you create. Unlike any of the other "fractal generators", it can be used to load and display previously created fractal pics, modify existing fractals, or create your own fractals. V1.1, binary only. Author: Doug Hawk.

### Fred Fish Disk 144

SoSubr

Scientific Subroutine Package from DEQU, ported to the Amiga to run with Aesoft Fortran. A valuable resource of mathematical and statistical source code for those doing Fortran work on the Amiga. Author: Unknown; ported to the Amiga by Glenn Everhart.

### Fred Fish Disk 145

Rim

Relational Information Manager, a full relational DBMS suitable for VERY large databases using B-Tree data storage, crude (by today's standards) user interface, but full source code is provided. RIM runs on a wide variety of systems, small and large, and produce compatible databases. Includes a built-in HELP database and a programming language. Full Fortran source code and documentation included. Author: Various, Amiga port by Glenn Everhart.

### Fred Fish Disk 146

AnalysCalc

V22-3D of Glenn Everhart's large and powerful spreadsheet program, update to FF104. Extra features "to have some pretensions of acting as an 'integrated system'". A virtual memory system supporting up to 18000 columns and 18000 rows, multiple equations per cell, an outlining system, built-in cell annotation, and datafile access from any cell(s) of the sheet, plus an array of functions not present in most commercial spreadsheets. Source and documentation in src'd form.

### Fred Fish Disk 147

Csh

Modification of csh like shell to provide file name completion and argument execution. Requires ARP 1.1. Binary only, but includes diffs for the reference 2.07 source base. Author: Matt Dillon; enhancements by John Widen.

### Fred Fish Disk 148

DMouse

Variable screen blanking, mouse blanker, auto window activator, mouse accelerator, popdi style programmable command key, pop window to front, push window to back, etc. widget. Very useful program! V1.06, includes source. Author: Matt Dillon.

### Fred Fish Disk 149

Net

Link protocol provides essentially an unlimited number of reliable connections between processes on two machines, where each can be either an Amiga or a Unix (BSD4.3) machine. Works on the Amiga with any EXEC device that looks like the serial device. Works on UNIX with tty and socket devices. Achieves better than 95% average throughput on file transfers. V1.20, includes source for both the Amiga and Unix versions. Author: Matt Dillon.

### Fred Fish Disk 150

Tab

Tablature writing program, with instruments for a banjo and string guitar. Binary only. Author: Jeff DeRenzo.

### TinyProlog

VT-PROLOG is a simple prolog interpreter provided with full source code to encourage experimentation with the PROLOG language and implementations. Version 1.1, includes source. Author: Bill and Bev Thompson.

### Fred Fish Disk 146

Blanker2

A screen blanking program that turns the screen black after 90 seconds of keyboard and mouse inactivity. V1.27.88, includes source. Author: Joe Hishens.

### Fred Fish Disk 147

C-Light

A demo copy of a commercial ray tracing program, identical to commercial version but limited to test objects per scene. Binary only. Author: Ronald Peterson.

### Fred Fish Disk 148

OrcLists

Complete CRC check files for FF129-141 and FF143-145 of the library, using the orc program from FF133. Made directly from Fred's master library. FF142 omitted due to a problem with the orc program. Author: Fred Fish.

### Fred Fish Disk 149

DmeMacros

A set of DME macros which utilize templates to turn DME into a language-sensitive editor for C, Pascal, Module-2, and Fortran. By Jerry Mack.

### Fred Fish Disk 150

Memopad

A shareware intuition-based memo reminder program. Neely done. V1.1, binary only. Author: Michael Griebling.

### Fred Fish Disk 151

To Be Continued.....

### In Conclusion

To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.

### IMPORTANT NOTICE!

This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial purposes only. Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.

Any non-commercial Amiga user group wishing to duplicate this list should contact:  
PIM Publications, Inc.  
P.O. Box 869  
Fall River, MA 02722

PIM Publications Inc. is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga.



# Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back issues, or ordering Amiga™ Public Domain Software

Name \_\_\_\_\_  
 Street \_\_\_\_\_  
 City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_  
 Amount Enclosed \_\_\_\_\_

Please circle the appropriate item:      **New Subscription**      **Renewal**

Please start my subscription to Amazing Computing™ with the next available issue  
 or renew my current subscription.

I have enclosed \$24.00 for 12 issues in the U.S. (\$36.00 Canada and Mexico, \$44.00 foreign surface).  
 All funds must be in U.S. Currency on a U.S. Bank

## Back Issues:

\$5.00 each US, \$6.00 each Canada and Mexico, \$7.00 each Foreign Surface.

Please circle your Back issue choices below:

Vol1.1	Vol1.2	Vol1.3	Vol1.4	Vol1.5	Vol1.6	Vol1.7	Vol1.8	Vol1.9	Vol2.1	Vol2.2	Vol2.3	Vol2.4	Vol2.5
Vol2.6	Vol2.7	Vol2.8	Vol2.9	Vol2.10	Vol2.11	Vol2.12	Vol3.1	Vol3.2	Vol3.3	Vol3.4	Vol3.5	Vol3.6	Vol3.7
Vol3.8													

## Public Domain Software:

\$6.00 each for subscribers (yes, even the new ones!)

\$7.00 each for non subscribers

(three disk minimum on all foreign orders)

Please circle your Public Domain Software choices below:

### Amazing on Disk:

AC#1 Source & Listings V3.8&V3.9

### Amicus:

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13
A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	A26

### Fred Fish:

FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13
FF14	FF15	FF16	FF17	FF18	FF19	FF20	FF21	FF22	FF23	FF24	FF25	FF26
FF27	FF28	FF29	FF30	FF31	FF32	FF33	FF34	FF35	FF36	FF37	FF38	FF39
FF40	FF41	FF42	FF43	FF44	FF45	FF46	FF47	FF48	FF49	FF50	FF51	FF52
FF53	FF54	FF55	FF56	FFNA	FF58	FF59	FF60	FF61	FF62	FF63	FF64	FF65
FF66	FF67	FF68	FF69	FF70	FF71	FF72	FF73	FF74	FF75	FF76	FF77	FF78
FF79	FFNA	FF81	FF82	FF83	FF84	FF85	FF86	FF87	FFNA	FF89	FF90	FF91
FF92	FF93	FF94	FF95	FF96	FF97	FF98	FF99	FF100	FF101	FF102	FF103	FF104
FF105	FF106	FF107	FF108	FF109	FF110	FF111	FF112	FF113	FF114	FF115	FF116	FF117
FF118	FF119	FF120	FF121	FF122	FF123	FF124	FF125	FF126	FF127	FF128	FF129	FF130
FF131	FF132	FF133	FF134	FF135	FF136	FF137	FF138	FF139	FF140	FF141	FF142	FF143
FF144	FF145	FF146										

(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

PiM Publications, Inc.  
 P.O.Box 869  
 Fall River, MA 02722

Please allow 4 to 6 weeks for delivery



# Amazing Computing Reader Service Card

Want to know more about Amazing Computing advertisers and their products? Here's your chance! Fill out the card below (for instructions, see page 96) and drop it in the mail (US only). We will process your request promptly and forward it to the advertiser.

Quick, easy, and efficient—just like all services of Amazing Computing: Your Original Amiga Monthly Resource.

FREE INFORMATION

## Amazing Computing Reader Service Card

Name \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_  
Telephone \_\_\_\_\_

September card Valid until 11/30/88  
Please see page 96 for instructions.

### A. Do you own an Amiga?

- ☐ 1. Amiga 1000  
☐ 2. Amiga 500  
☐ 3. Amiga 2000

- ☐ 4. Soon  
☐ 5. Not Yet  
☐ 6. Just Looking

### B. What Amiga hardware product do you plan to buy next?

- ☐ 1. Amiga 500  
☐ 2. Amiga 2000  
☐ 3. Memory Expansion  
☐ 4. Hard Drive  
☐ 5. IBM Emulators (idecator or Bridgeboard)

- ☐ 6. Printer  
☐ 7. Modem  
☐ 8. Music Tool  
☐ 9. Video Product  
☐ 10. Other

### C. What Amiga software product do you plan to buy next?

- ☐ 1. C' Language  
☐ 2. Forth Language  
☐ 3. Module-2 Language  
☐ 4. Assembly Language  
☐ 5. BASIC Language  
☐ 6. Entertainment  
☐ 7. Telecommunications

- ☐ 8. Spreadsheet  
☐ 9. Database  
☐ 10. Financial  
☐ 11. Video  
☐ 12. Graphics  
☐ 13. Music  
☐ 14. Other

### D. Where do you buy your Amiga products?

- ☐ 1. Local Amiga Dealer  
☐ 2. Discount Department Store

- ☐ 3. Manufacturer  
☐ 4. Mail Order

### E. Which type of articles do you like to see in Amazing Computing?

- ☐ 1. C' Language  
☐ 2. Forth Language  
☐ 3. Module-2 Language  
☐ 4. Assembly Language  
☐ 5. BASIC Language  
☐ 6. Game Reviews  
☐ 7. Business Reviews  
☐ 8. Hardware Product Reviews  
☐ 9. Software Product Reviews

- ☐ 8. Programming How To's  
☐ 10. Business How To's  
☐ 11. Video Articles  
☐ 12. Graphics Articles  
☐ 13. Music Articles  
☐ 14. Hardware How To's  
☐ 15. PDS Updates  
☐ 16. Interviews  
☐ 17. Other

### F. Which articles would you like to see more of in Amazing Computing?

- ☐ 1. C' Language  
☐ 2. Forth Language  
☐ 3. Module-2 Language  
☐ 4. Assembly Language  
☐ 5. BASIC Language  
☐ 6. Game Reviews  
☐ 7. Business Reviews  
☐ 8. Hardware Product Reviews  
☐ 9. Software Product Reviews

- ☐ 8. Programming How To's  
☐ 10. Business How To's  
☐ 11. Video Articles  
☐ 12. Graphics Articles  
☐ 13. Music Articles  
☐ 14. Hardware How To's  
☐ 15. PDS Updates  
☐ 16. Interviews  
☐ 17. Other

### G. Are you a subscriber to Amazing Computing?

- ☐ 1. Yes  
☐ 2. No

101	102	103	104	105	201	202	203	204	205
106	107	108	109	110	206	207	208	209	210
111	112	113	114	115	211	212	213	214	215
116	117	118	119	120	216	217	218	219	220
121	122	123	124	125	221	222	223	224	225
126	127	128	129	130	226	227	228	229	230
131	132	133	134	135	231	232	233	234	235
136	137	138	139	140	236	237	238	239	240
141	142	143	144	145	241	242	243	244	245
146	147	148	149	150	246	247	248	249	250
151	152	153	154	155	251	252	253	254	255
156	157	158	159	160	256	257	258	259	260
161	162	163	164	165	261	262	263	264	265
166	167	168	169	170	266	267	268	269	270
171	172	173	174	175	271	272	273	274	275
176	177	178	179	180	276	277	278	279	280
181	182	183	184	185	281	282	283	284	285
186	187	188	189	190	286	287	288	289	290
191	192	193	194	195	291	292	293	294	295
196	197	198	199	200	296	297	298	299	300

101

## Amazing Computing Reader Service Card

Name \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_  
Telephone \_\_\_\_\_

September card Valid until 11/30/88  
Please see page 96 for instructions.

### A. Do you own an Amiga?

- ☐ 1. Amiga 1000  
☐ 2. Amiga 500  
☐ 3. Amiga 2000

- ☐ 4. Soon  
☐ 5. Not Yet  
☐ 6. Just Looking

### B. What Amiga hardware product do you plan to buy next?

- ☐ 1. Amiga 500  
☐ 2. Amiga 2000  
☐ 3. Memory Expansion  
☐ 4. Hard Drive  
☐ 5. IBM Emulators (idecator or Bridgeboard)

- ☐ 6. Printer  
☐ 7. Modem  
☐ 8. Music Tool  
☐ 9. Video Product  
☐ 10. Other

### C. What Amiga software product do you plan to buy next?

- ☐ 1. C' Language  
☐ 2. Forth Language  
☐ 3. Module-2 Language  
☐ 4. Assembly Language  
☐ 5. BASIC Language  
☐ 6. Entertainment  
☐ 7. Telecommunications

- ☐ 8. Spreadsheet  
☐ 9. Database  
☐ 10. Financial  
☐ 11. Video  
☐ 12. Graphics  
☐ 13. Music  
☐ 14. Other

### D. Where do you buy your Amiga products?

- ☐ 1. Local Amiga Dealer  
☐ 2. Discount Department Store

- ☐ 3. Manufacturer  
☐ 4. Mail Order

### E. Which type of articles do you like to see in Amazing Computing?

- ☐ 1. C' Language  
☐ 2. Forth Language  
☐ 3. Module-2 Language  
☐ 4. Assembly Language  
☐ 5. BASIC Language  
☐ 6. Game Reviews  
☐ 7. Business Reviews  
☐ 8. Hardware Product Reviews  
☐ 9. Software Product Reviews

- ☐ 8. Programming How To's  
☐ 10. Business How To's  
☐ 11. Video Articles  
☐ 12. Graphics Articles  
☐ 13. Music Articles  
☐ 14. Hardware How To's  
☐ 15. PDS Updates  
☐ 16. Interviews  
☐ 17. Other

### F. Which articles would you like to see more of in Amazing Computing?

- ☐ 1. C' Language  
☐ 2. Forth Language  
☐ 3. Module-2 Language  
☐ 4. Assembly Language  
☐ 5. BASIC Language  
☐ 6. Game Reviews  
☐ 7. Business Reviews  
☐ 8. Hardware Product Reviews  
☐ 9. Software Product Reviews

- ☐ 8. Programming How To's  
☐ 10. Business How To's  
☐ 11. Video Articles  
☐ 12. Graphics Articles  
☐ 13. Music Articles  
☐ 14. Hardware How To's  
☐ 15. PDS Updates  
☐ 16. Interviews  
☐ 17. Other

### G. Are you a subscriber to Amazing Computing?

- ☐ 1. Yes  
☐ 2. No

101	102	103	104	105	201	202	203	204	205
106	107	108	109	110	206	207	208	209	210
111	112	113	114	115	211	212	213	214	215
116	117	118	119	120	216	217	218	219	220
121	122	123	124	125	221	222	223	224	225
126	127	128	129	130	226	227	228	229	230
131	132	133	134	135	231	232	233	234	235
136	137	138	139	140	236	237	238	239	240
141	142	143	144	145	241	242	243	244	245
146	147	148	149	150	246	247	248	249	250
151	152	153	154	155	251	252	253	254	255
156	157	158	159	160	256	257	258	259	260
161	162	163	164	165	261	262	263	264	265
166	167	168	169	170	266	267	268	269	270
171	172	173	174	175	271	272	273	274	275
176	177	178	179	180	276	277	278	279	280
181	182	183	184	185	281	282	283	284	285
186	187	188	189	190	286	287	288	289	290
191	192	193	194	195	291	292	293	294	295
196	197	198	199	200	296	297	298	299	300

102

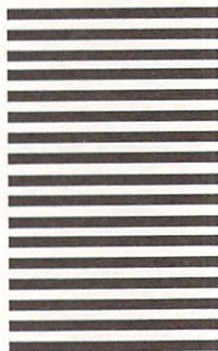


# FREE INFORMATION

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 36 FALL RIVER, MA

POSTAGE WILL BE PAID BY ADDRESSEE

**Amazing**  
COMPUTINGP.O. Box 869  
Fall River, MA 02722-0869NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 36 FALL RIVER, MA

POSTAGE WILL BE PAID BY ADDRESSEE

**Amazing**  
COMPUTINGP.O. Box 869  
Fall River, MA 02722-0869NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES



## Do It NOW!!!

Subscribe TODAY and save over 49% on Amazing Computing!

*Amazing Computing is available to United States subscribers at \$24.00 for 12 issues. That's a savings of \$23.40 off the newsstand price. Canada and Mexico, \$36.00. Foreign Surface to all other countries, \$44.00.  
(Air mail and first class rates available upon request.)*

Enter my 12 issue subscription today!

Name \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_

Amount Enclosed \_\_\_\_\_

Sorry, we cannot accept credit cards or requests for billing at this time.  
All funds must be in U.S. Currency drawn on a U.S. Bank.  
Please allow four to six weeks for processing.

## Save Time and Money!

Subscribe TODAY and save over 49% on Amazing Computing!

*Amazing Computing is available to United States subscribers at \$24.00 for 12 issues. That's a savings of \$23.40 off the newsstand price. Canada and Mexico, \$36.00. Foreign Surface to all other countries, \$44.00.  
(Air mail and first class rates available upon request.)*

Enter my 12 issue subscription today!

Name \_\_\_\_\_  
Street \_\_\_\_\_  
City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_

Amount Enclosed \_\_\_\_\_

Sorry, we cannot accept credit cards or requests for billing at this time.  
All funds must be in U.S. Currency drawn on a U.S. Bank.  
Please allow four to six weeks for processing.

## Special Subscription Bonus

*See the special offer on the inside rear cover of this issue*

**Clip and \$AVE !!**  
(over 49% of the newsstand price!)



Please Return to:

**Amazing**  
COMPUTING

P.O. Box 869

Fall River, MA 02722-0869

Place this order form in an envelope with your  
Check or Money Order.

**Clip and \$AVE !!**

(over 49% of the newsstand price!)

Please Return to:

**Amazing**  
COMPUTING

P.O. Box 869

Fall River, MA 02722-0869

Place this order form in an envelope with your  
Check or Money Order.

**Special Subscription Bonus**

*See the special offer on the inside rear cover of this issue*



# ***I Will NOT Miss Out!!!***

**I will subscribe TODAY and save over 49% on Amazing Computing,  
plus receive my Amazing Product Guide: Winter '89.**

*I can't pass up an offer like this! Enclosed is my check for \$24 for a one-year  
subscription to Amazing Computing. Canada and Mexico, \$36.00. Foreign  
Surface to all other countries, \$44.00.*

*(Air mail and first class rates available upon request.)*

*Enter my 12 issue subscription today!*

Name \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_

Amount Enclosed \_\_\_\_\_

Sorry, we cannot accept credit cards or requests for billing at this time.

All funds must be in U.S. Currency drawn on a U.S. Bank.

Please allow four to six weeks for processing. We are not responsible for late entries.



**Please Return to:**

***Amazing***  
COMPUTING

P.O. Box 869

Fall River, MA 02722-0869

Place this order form in an envelope with your  
Check or Money Order.



# *Don't Miss An Issue!*

**Subscribe TODAY and save over 49% on Amazing Computing!**

*Amazing Computing is available to United States subscribers at \$24.00 for 12 issues. That's a savings of \$23.40 off the newsstand price. Canada and Mexico, \$36.00. Foreign Surface to all other countries, \$44.00.  
(Air mail and first class rates available upon request.)*

*Enter my 12 issue subscription today!*

Name \_\_\_\_\_

Street \_\_\_\_\_

City \_\_\_\_\_ St. \_\_\_\_\_ Zip \_\_\_\_\_

Amount Enclosed \_\_\_\_\_

Sorry, we cannot accept credit cards or requests for billing at this time.  
All funds must be in U.S. Currency drawn on a U.S. Bank.  
Please allow four to six weeks for processing.



Please Return to:

*Amazing*  
COMPUTING

P.O. Box 869

Fall River, MA 02722-0869

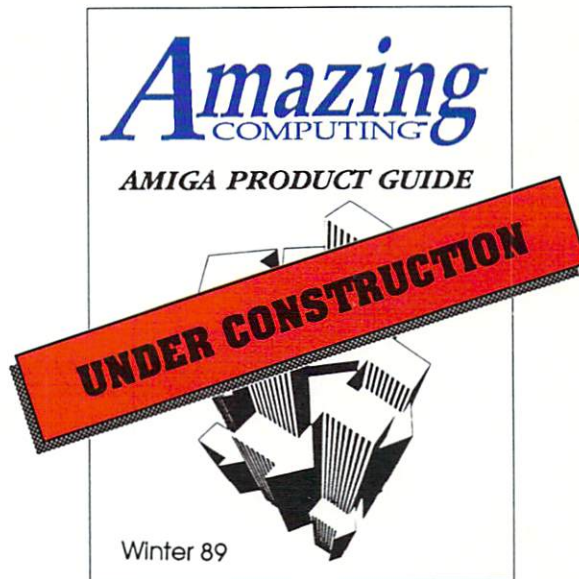
Place this order form in an envelope with your  
Check or Money Order.



# Don't Miss Out!

---

*Subscribe now and receive the Amazing Product Guide: Winter '89  
absolutely FREE!*



*The Amazing Product Guide: Winter '89* will be a complete descriptive listing of *all* products available for the Amiga. If you have appreciated the theme-specific product guide installments in past few issues of AC, just imagine bigger and better chunks of information. The *Amazing Product Guide: Winter '89* will feature longer, more detailed descriptions and even more products!

The Guide will be available on newsstands in December at the standard cover price of \$3.95. Subscribers will receive the Guide along with their December AC for no additional charge! The \$24 that usually gets you a year full of great Amiga information through Amazing Computing, can now bring you that same year of information-packed AC *and* the *Amazing Product Guide: Winter '89*.

## ***Don't Miss Out!***

The *Amazing Product Guide: Winter '89* will be a must for all Amiga users, but your Amazing Dealer will sellout quickly! Guarantee your *Amazing Product Guide: Winter '89* by mailing your subscription request today!

## ***Order Early!***

Your subscription request **MUST** be received in time for our mailing. NO exceptions! Mail TODAY!

---



**New!**  
Version 3.0  
includes overscan,  
half-brite support and more!

# DIGI VIEW

## DELIVERS ULTIMATE GRAPHICS POWER

Bring the world into your Amiga with Digi-View, the 4096 color video digitizer. In seconds you can capture any photograph or object your video camera can see in full color and with clarity never before available on a home computer. Digi-View's advanced features include:

- Dithering routines give up to 100,000 apparent colors on screen
- NewTek's exclusive Enhanced Hold-and-Modify mode allows for exceptionally detailed images
- Digitize images in any number of colors from 2 to 4096
- Print, animate, transmit, store, or manipulate images with available IFF compatible programs
- Digitize in all Amiga resolution modes (320x200, 320x400, 640x200, 640x400)

**"Digi-View sets new standards for graphics hardware" - InfoWorld**

Digi-View is available now at your local Amiga dealer or call:  
1-800-843-8934

**ONLY \$199.95**



**NewTek**  
INCORPORATED

All photos actual unretouched Digi-View pictures shot directly off the 1080 Amiga monitor.